ITERATIVE EQUALIZATION AND DECODING USING REDUCED-STATE SEQUENCE ESTIMATION BASED SOFT-OUTPUT ALGORITHMS

A Thesis

by

RAJA VENKATESH TAMMA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2003

Major Subject: Electrical Engineering

ITERATIVE EQUALIZATION AND DECODING USING REDUCED-STATE SEQUENCE ESTIMATION BASED SOFT-OUTPUT ALGORITHMS

A Thesis

by

RAJA VENKATESH TAMMA

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Krishna R. Narayanan (Chair of Committee)	
Scott Miller (Member)	Gwan Choi (Member)
Jyh-Charn Liu (Member)	Chanan Singh (Head of Department)

May 2003

Major Subject: Electrical Engineering

ABSTRACT

Iterative Equalization and Decoding Using Reduced-State
Sequence Estimation Based Soft-Output Algorithms. (May 2003)
Raja Venkatesh Tamma, B.E. (Hons.),
Birla Institute of Technology & Science, Pilani
Chair of Advisory Committee: Dr. Krishna R. Narayanan

We study and analyze the performance of iterative equalization and decoding (IED) using an M-BCJR equalizer. We use bit error rate (BER), frame error rate simulations and extrinsic information transfer (EXIT) charts to study and compare the performances of M-BCJR and BCJR equalizers on precoded and non-precoded channels. Using EXIT charts, the achievable channel capacities with IED using the BCJR, M-BCJR and MMSE LE equalizers are also compared. We predict the BER performance of IED using the M-BCJR equalizer from EXIT charts and explain the discrepancy between the observed and predicted performances by showing that the extrinsic outputs of the M-BCJR algorithm are not true logarithmic-likelihood ratios (LLR's). We show that the true LLR's can be estimated if the conditional distributions of the extrinsic outputs are known and finally we design a practical estimator for computing the true LLR's from the extrinsic outputs of the M-BCJR equalizer.

To my Parents

ACKNOWLEDGMENTS

I wish to thank my advisor Dr. Krishna Narayanan for giving me an opportunity to work with him. He has been extremely helpful and played a central role in guiding my thesis efforts. The several discussions that I had with him contributed richly to the progress of my thesis work.

I would also like to thank the other members of my committee, Dr. S. Miller, Dr. G. Choi and Dr. J. C. Liu for their cooperation and suggestions throughout my program.

Lastly, I take this opportutnity to thank my parents and brother for their unending love, support and guidance.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Iterative equalization and decoding (IED) is an offshoot of the increasing interest within the channel coding community in iterative decoding techniques [1]. It was first proposed by C. Douillard *et al.* [2] to minimize the bit error rate (BER) by modeling the ISI channel as a convolutional code and applying iterative decoding on the serial concatenated code (SCC) structure thus realized in conjunction with an outer error-correction code (ECC).

In a typical digital communications system, the data is protected by an ECC and the coded bits are modulated and transmitted over the ISI channel. In the receiver, the channel symbols are demodulated and decoded to recover the user data. The transmission channel and the demodulator may be equivalently represented by a discrete-time transversal filter (DTTF). The structure of a DTTF resembles a conventional convolutional code. Thus, the input symbols to the decoder may equivalently be considered as the output symbols of a convolutional code cascaded with an DTTF model to which additive white Gaussian noise (AWGN) has been added. If an interleaver is inserted between the ECC and the DTTF, an SCC results. In the iterative decoder, the interleaver decorrelates the feedback information exchanged between the equalizer and the decoder thus making them appear as independent decoding modules. This is essential for achieving good performance in IED.

In IED, a convolutional code is typically chosen as the ECC. This choice is motivated by the relative ease with which a soft-input soft-output (SISO) decoder can be implemented for a convolutional code in comparison with a block code. Thus,

---

The journal model is *IEEE Transactions on Automatic Control.*

the convolutional code forms the outer code and the DTTF forms the inner code of the serial concatenated convolutional code (SCCC) [3]. We may use algorithms based on maximum *a posteriori* probability (MAP) sequence estimation such as the Soft-Output Viterbi Algorithm (SOVA) [4] or MAP symbol estimation *e.g.,* Bahl-Cocke-Jelinek-Raviv (BCJR) [5] for obtaining soft-information in IED. It was shown in [3] that significant performance gains can be obtained from SCCC if and only if the inner code is recursive. To make the DTTF appear recursive to the IED device, we use a precoder [6] at the input of the DTTF. When the DTTF is made recursive through such precoding, an interleaving gain results and the BER performance improves significantly.

IED, as other iterative decoding algorithms do, suffers from high computational complexity. There has been a welter of research in the coding community aimed at reducing it at the expense of degraded performance. In [6], it was shown that convolutional codes with small constraint lengths outperform longer constraint lengths codes at low signal-to-noise ratios (SNR's). Therefore, the equalizer complexity is the real bottleneck in IED. The need for reduced-complexity equalizer algorithms assumes even greater importance in the presence of channels with long impulse responses. Our research efforts are geared towards exploring the performance of such reduced-complexity soft-output algorithms for equalization on Extended Partial Response (EPR) channels and analyzing their performance. We focussed on the BCJR algorithm and its M-variant [7]. Although there are several published simulation results showing the performance of IED with an M-BCJR equalizer, several questions about its performance remained unanswered.

In this thesis, we focus on understanding the behavior of the M-BCJR equalizer in IED and compare its performance with other reduced-complexity equalizers. The ECC is restricted to be a convolutional code and its corresponding decoder in the

IED module is always restricted to use the BCJR algorithm.

The remainder of the report is organized as follows. We discuss the significance of IED and give a summary of past research in this area in Chapter II. In Chapter III, we review the BCJR algorithm, its M-variant and the extrinsic information transfer (EXIT) charts [8]. In Chapter IV, we use BER and FER simulations and EXIT charts to study the M-BCJR equalizer performance on precoded and non-precoded channels. Linear equalizers (*e.g.*, LE's) and DFE's [9] whose filter parameters are updated based on criteria such as minimum-mean-squared-error (MMSE) are studied extensively in the literature [10], [11], [12], [13]. We compare the losses in achievable channel capacities of IED using the M-BCJR and MMSE LE equalizers measured with reference to the BCJR equalizer. It is also interesting to note that unlike an M-BCJR equalizer, an M-BCJR decoder suffers severe degradation when used on a simple convolutional code. We try to explain this contrasting behavior of M-BCJR using empirical arguments and with the help of simulations. We present simulation results for IED with an M-BCJR equalizer on precoded channels also. Unlike the non-precoded channel case, our simulations showed that the IED performance loss with an M-BCJR equalizer on a precoded channel was significant for small values of M.

In Chapter V, we see how close the simulated BER performance of the equalizer is to the threshold predicted from the EXIT charts. When the BCJR equalizer was used, the BER simulations for IED agreed with the expected threshold on both non-precoded and precoded channels. On the other hand, when the M-BCJR equalizer was used, there was a considerable gap between the observed BER performance and the expected convergence thresholds on precoded channels especially for small values of M. On non-precoded channels, the discrepancy between the actual performance and the expected convergence threshold using an M-BCJR equalizer was insignificant

at BER's of practical interest. To explain this behavior, we show that the extrinsic logarithmic likelihood ratios (LLR's) at the output of the M-BCJR equalizer are not true LLR's. Hence the traditional practice of directly using them as feedback to the outer decoder is sub-optimal. We show that the true values of the extrinsic LLR's can be obtained through Bayesian estimation given a conditional distribution of the output extrinsic LLR's for each symbol *viz.*, -1 and $+1$ in the input alphabet of the DTTF. We also present simulation results for such a *genie-assisted* IED to show that the actual BER performance agrees with the predicted threshold.

In Chapter VI we develop a practical algorithm to estimate the true extrinsic LLR's at the equalizer output by estimating the pdf's of the output extrinsic LLR's. It is interesting to note that unlike a non-precoded channel, -1's and $+1$'s are unequally protected when precoding is used. Thus, this task assumes different forms for the non-precoded and precoded channels. However, we try to design an estimator only for the precoded channel. This is because in most practical applications, a convolutional code is used as the ECC and in such cases, the performance of the M-BCJR equalizer on a non-precoded channel is satisfactory and agrees with the predicted performance.

The conclusions are given in Chapter VII.

CHAPTER II

ITERATIVE EQUALIZATION AND DECODING

Iterative equalization and decoding is employed in the presence of an ISI channel to counter the effects of ISI and minimize the bit error rate (BER). It was pioneered by C. Douillard *et al.* [2] as an application of the iterative decoding principle [1], [14] and has been an object of extensive research henceforth.

A. System Model

A typical communication system is shown in Figure 1. The data is protected by an ECC. The coded bits c_k are mapped to a symbol from the alphabet χ of the signal constellation and transmitted over an ISI channel. For simplicity, we assume binary phase shift keying (BPSK) modulation, i.e., $\chi = \{-1, +1\}$. In the receiver, the channel output z_k is passed through a whitened matched filter (WMF) and sampled every T seconds, the symbol duration. The cascade of the ISI channel with the WMF and the sampler may be equivalently treated as a DTTF. The DTTF is a finite impulse response (FIR) filter and can be represented by an L-tap delay line with coefficients $f_m, m = 0, 1, 2, \ldots, L-1$. Thus the output of the DTTF at time k may be written as

$$w_k = \sum_{m=0}^{L-1} f_m x_{k-m} + n_k$$

where n_k is a sample of a white Gaussian noise process at time k. We shall refer to a communication system model in which the ISI channel, the WMF and the sampler are replaced with the DTTF as its DTTF model.

From the analogy with SCCC's [3], we know that the BER can be significantly improved by iteratively passing soft information rather than hard symbol estimates

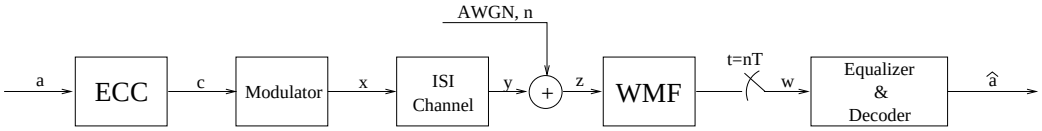

Fig. 1. A typical communication system

between the equalizer and the decoder. However, the computational complexity of such a system increases with the number of iterations and the channel impulse response length L. To facilitate IED, a convolutional code is used as the ECC. It is possible to obtain large interleaving gains in SCCC's if the inner code (DTTF) is recursive [3]. To make the DTTF appear recursive, we may use a precoder as in [6].

The DTTF model of the communication system is shown in Figure 2. A binary data stream, $a_k,\ k = 1, 2, \ldots, N$ of length N is encoded by the convolutional encoder generating a multiplexed stream of coded bits, $c_k,\ k = 1, 2, \ldots, K$ of length $K = (1/R) \cdot N$ where R is the rate of the convolutional code. The coded bits c_k are pseudo-randomly permuted by the interleaver $\pi(\cdot)$. The interleaved stream, $\pi(c_k),\ k = 1, 2, \ldots, K$ is binary-phase shift keying (BPSK) modulated to obtain the sequence $x_k,\ k = 1, 2, \ldots, K,\ x_k \in \chi$ which is then convolutionally encoded by the DTTF to produce $y_k,\ k = 1, 2, \ldots, K,\ y_k \in \zeta$ where ζ is the output alphabet of the DTTF when the input alphabet is χ. Since the code rate of DTTF is unity, the length of the output stream is equal to K.

B. Principle of IED

IED is based on the principle of iterative decoding applied to SCCC's [15]. The receiver structure for IED is depicted in Figure 3. Both the equalizer and the decoder are SISO devices. The equalizer computes the *a posteriori* probabilities (APP's), $P(x_k = x | z_1, z_2, \ldots, z_K),\ x \in \chi,\ k = 1, 2, \ldots, K$, given K received symbols $z_k,\ k =$

$1, 2, \ldots, K$ and outputs the extrinsic LLR's, $L^I(x_k),\ k = 1, 2, \ldots, K$ defined as the *a posteriori* LLR minus the *a priori* LLR. The superscript I refers to the inner SISO module *viz.,* the equalizer.

$$L^I(x_k) = \ln\left(\frac{P(x_k = +1|z_1, z_2, \ldots, z_K)}{P(x_k = -1|z_1, z_2, \ldots, z_K)}\right) - \ln\left(\frac{P(x_k = +1)}{P(x_k = -1)}\right). \tag{2.1}$$

The *a priori* LLR, $L(x_k),\ k = 1, 2, \ldots, K$ represents the *a priori* information about the probability that $x_k \in \chi,\ k = 1, 2, \ldots, K$ assumes a particular value. They are provided by the APP decoder. In the first equalization step, the *a priori* information, $L(x_k),\ k = 1, 2, \ldots, K$ is not available and all values are assumed to be equally probable i.e., $L(x_k) = 0,\ \forall\ k$. The *a priori* information for the decoder is obtained by deinterleaving the output extrinsic LLR's of the equalizer, $L(c_k) = \pi^{-1}(L^I(x_k))$. Similar to the equalizer, the decoder also computes the APP's $P(c_k = c|L(c_1), L(c_2), \ldots, L(c_K)),\ c \in \{0, 1\}$ given the K code bit LLR's $L(c_k),\ k = 1, 2, \ldots, K$ and outputs the extrinsic LLR's, $L^O(c_k),\ k = 1, 2, \ldots, K$ defined as the output LLR minus the *a priori* LLR. The superscipt O refers to the outer decoder.

$$L^O(x_k) = \ln\left(\frac{P(c_k = +1|L(c_1), L(c_2), \ldots, L(c_K))}{P(c_k = -1|L(c_1), L(c_2), \ldots, L(c_K))}\right) - \ln\left(\frac{P(c_k = +1)}{P(c_k = -1)}\right). \tag{2.2}$$

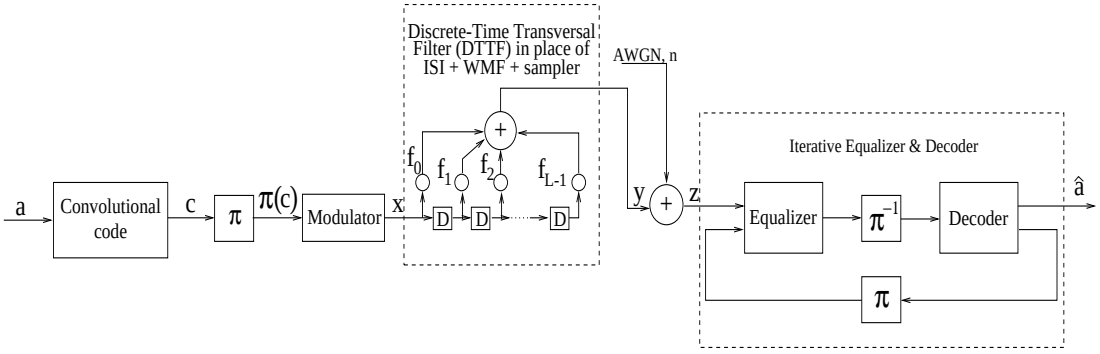

Fig. 2. DTTF model of a communication system

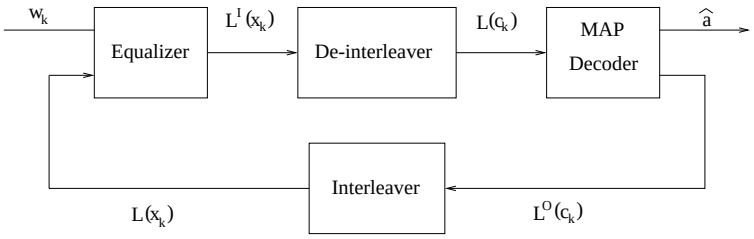

Fig. 3. Receiver model for IED

The extrinsic LLR's of the decoder are interleaved to obtain the intrinsic information for the equalizer i.e., $L(x_k) = \pi(L^O(c_k)),\ k = 1, 2, \ldots, K$. The decoder also outputs estimates of the data bits as

$$\hat{a_k} = \underset{b\in\{0,1\}}{arg\,max} \Big\{ P(b_k = b | L(c_1), L(c_2), \ldots, L(c_K)) \Big\}. \tag{2.3}$$

This process is repeated several times over a block of received symbols. The BCJR algorithm and its M-variant as used in IED are described in Chapter III.

C. Summary of Past Research

Linear equalization (LE) and decision feedback equalization (DFE) [9], [16] (non-linear) are sub-optimal equalization techniques. Their filter parameters may be selected on the basis of a variety of optimization criteria such as minimum-mean-squared-error (MMSE) and the least-mean-squares (LMS) algorithm. On the other hand, equalization techniques based on MLSE/MAP are optimal. The SOVA [4] and BCJR [5] are examples of equalization algorithms based on MAP sequence estimation and symbol estimation algorithms respectively. However, they are faced with the disadvantage of exponentially increasing complexity as the length of channel impulse response increases. Joint equalization and decoding is the optimal detection procedure but is not a viable solution owing to its exceptionally high complexity.

IED was first proposed in [2] and was further developed in [17], [18] besides several other papers. In view of the high decoding complexity associated with the MLSE/MAP equalizers, there have been several research efforts directed at developing reduced-complexity equalizers. A common technique to decrease the complexity of non-linear equalizers is to use reduced-search algorithms such as the M-BCJR, T-BCJR [7], [19], M-SOVA and T-SOVA [6] algorithms. Another approach is to use equalization algorithms in which the filter coefficients may be selected on the basis of different optimization criteria such as zero forcing (ZF) or MMSE [9]. Although such algorithms are sub-optimal, they offer the advantage of quadratically increasing complexity as the length of the channel impulse response increases. In [10], the MAP equalizer in the IED framework is replaced with a soft-interference canceller based on linear filters with a very low computational complexity. The filter coefficients are updated using a least-mean-square (LMS) algorithm. This idea was further developed in [11] where the filter coefficients were obtained by the LMS algorithm to match the output of a MAP equalizer. For various signal-to-noise ratios (SNR's) and feedback information constellations, a linear estimate of the MAP equalizer is stored in a table and used for equalization in the receiver. A similar approach was followed in [12] but a known channel impulse response (as occurring in magnetic recording applications) was assumed. The equalizer filter output was assigned a reliability measure to enable the receiver to decide whether the linear algorithm should be used instead of the MAP algorithm. In [13], a general structure for a SISO device based on MMSE equalization was described and different implementations based on LE and DFE were derived. It analyzes the performance of the equalizer and the decoder and gives suggestions on how to select appropriate codes and SISO equalizers.

CHAPTER III

REVIEW OF BCJR ALGORITHM AND EXIT CHARTS

The Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm is a SISO processing tool. In general, it is applicable to any Markov chain and produces *a posteriori* probabilities. Since we are interested in the application of the BCJR algorithm to IED, we shall present it as adapted to convolutional codes.

IED consists of two SISO devices *viz.,* the equalizer and the decoder. They are based on the general BCJR algorithm presented in Section A and can be readily used as building blocks in any iterative decoder for convolutional codes. In Section B, we describe the M-BCJR algorithm.

EXIT charts [8] may be used to study the convergence behavior of an iterative decoder. The theory of EXIT charts and their utility are discussed in Section C.

A. SISO Module Based on the BCJR Algorithm

The SISO module shown in Figure 4 is a four-port device that accepts at the input the sequences of LLR's of the information bits, $\lambda^I(u)$ and the coded bits, $\lambda^I(c)$ and produces at the output sequences of *extrinsic* LLR's for the information bits, $\lambda^O(u)$ and the coded bits, $\lambda^O(c)$. Its input-output relationships and internal operations are given in subsection 1.

In our description of the BCJR algorithm, we assume that the encoder trellis is time-invariant. This is a valid assumption because we are dealing with the time-invariant trellises of the DTTF and convolutional codes in IED. For notation, we use the trellis section of the encoder trellis shown in Figure 5. The symbol e denotes a trellis edge starting from state s_e^α and ending at state s_e^β. u_e and c_e are the respective information and the code symbols associated with the edge e.

The SISO module can operate at *bit* level or *symbol* level. Quite often, the interleaver operates at *bit* level for improved performance. This necessitates the transformation of *symbol* LLR's into *bit* LLR's and *vice versa* if the SISO module is implemented at the *symbol* level (see subsection 2). Here, we describe the *symbol* level SISO module.

1. The Input-Output Relationships of the SISO Module

Assume that the information and code symbols are defined over a finite time index set [1,2, ... ,K]. Let the operator $\overset{*}{\log}$ be defined as

$$\overset{*}{\log_j}(a_j) \triangleq \log\left[\sum_{j=1}^{J} e^{a_j}\right] \tag{3.1}$$

From the input LLR's, $\lambda_k^I(u)$ and $\lambda_k^I(c),\ k = 1, 2, \ldots, K$, the output *extrinsic* LLR's, $\lambda_k^O(u)$ and $\lambda_k^O(c),\ k = 1, 2, \ldots, K$ are calculated as

$$\lambda_k^O(u) = \overset{*}{\log_{e:u_e=u}}\left\{\alpha_{k-1}(s_e^\alpha) + \lambda_k^I(c_e) + \beta_k(s_e^\beta)\right\} \tag{3.2}$$

$$\lambda_k^O(c) = \overset{*}{\log_{e:c_e=c}}\left\{\alpha_{k-1}(s_e^\alpha) + \lambda_k^I(u_e) + \beta_k(s_e^\beta)\right\} \tag{3.3}$$

The LLR's calculated according to (3.2) and (3.3) are termed *extrinsic* due to the fact that the computation of $\lambda_k^O(u)$ (and of $\lambda_k^O(u)$) does not depend on the corresponding input LLR $\lambda_k^I(u)$ (and $\lambda_k^I(u)$) and so it can be considered as an update

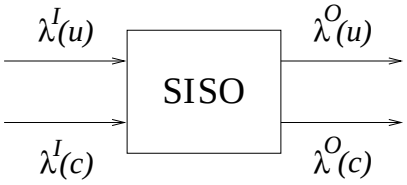

Fig. 4. Block diagram of a SISO module

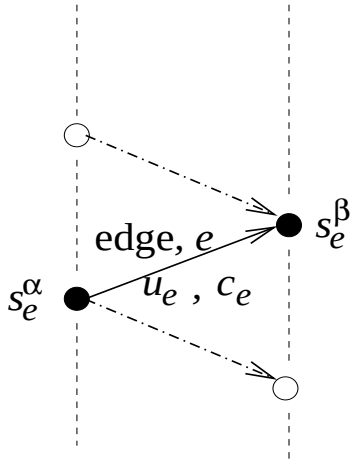

Fig. 5. Encoder trellis section defining notation for description of the SISO algorithm

of the input LLR based on the code constraints and the information provided by all homologous symbols in the sequence except the one corresponding to the same symbol interval.

The quantities $\alpha_k(\cdot)$ and $\beta_k(\cdot)$ in (3.2) and (3.3) are obtained through *forward* and *backward* recursions, respectively, as

$$\alpha_k(s) = \frac{\overset{*}{\underset{e:s_e^\beta=s}{\log}} \left\{\alpha_{k-1}(s_e^\alpha) + \lambda_k^I(u_e) + \lambda_k^I(c_e)\right\}}{\overset{*}{\underset{s}{\log}} \left\{ \overset{*}{\underset{e:s_e^\beta=s}{\log}} \left\{\alpha_{k-1}(s_e^\alpha) + \lambda_k^I(u_e) + \lambda_k^I(c_e)\right\} \right\}}, k = 1, 2, \ldots, K-1. \tag{3.4}$$

$$\beta_k(s) = \frac{\overset{*}{\underset{e:s_e^\alpha=s}{\log}} \left\{\beta_{k+1}(s_e^\beta) + \lambda_{k+1}^I(u_e) + \lambda_{k+1}^I(c_e)\right\}}{\overset{*}{\underset{s}{\log}} \left\{ \overset{*}{\underset{e:s_e^\beta=s}{\log}} \left\{\alpha_k(s_e^\alpha) + \lambda_{k+1}^I(u_e) + \lambda_{k+1}^I(c_e)\right\} \right\}}, k = 1, 2, \ldots, K-1. \tag{3.5}$$

with initial values

$$\alpha_0(s) = \begin{cases} 0, & s = S_0 \\ -\infty, & \text{otherwise.} \end{cases}$$

$$\beta_K(s) = \begin{cases} 0, & s = S_K \\ -\infty, & \text{otherwise.} \end{cases}$$

assuming that the encoder starts and ends in state 0. If the encoder ends in an unknown state, the initial values for the backward recursion are chosen as

$$\beta_K(s) = \alpha_K(s) \qquad \forall s$$

The denominators in (3.4) and (3.5) are normalization terms which help avoid numerical problems arising out of finite precision. The $\overset{*}{\log}$ operator may be simplified as

$$\overset{*}{\log_j}(a_j) = \max_j(a_j) + \delta(a_1, a_2, \ldots, a_J) \tag{3.6}$$

where $\delta(a_1, a_2, \ldots, a_J)$ is a correction term that can be computed recursively using a single-entry lookup table [20]. This simplification significantly decreases the computational complexity of the BCJR algorithm at the expense of a slight performance degradation.

2. Inter-conversion between *symbol* and *bit* Level LLR's

Inter-conversion operations between *symbol* and *bit* Level LLR's are necessitated by the presence of a bit-interleaver. These operations assume that the bits forming a symbol are independent. Suppose $\mathbf{u} = [u_1, u_2, \ldots, u_m]$ is a symbol formed by m bits. The extrinsic LLR λ_j of the jth bit u_j within the symbol $\mathbf{u}$ is obtained as

$$\lambda_j^O(u) = \overset{*}{\underset{\mathbf{u}:u_j=1}{\log}} [\lambda^O(\mathbf{u}) + \lambda^I(\mathbf{u})] - \overset{*}{\underset{\mathbf{u}:u_j=0}{\log}} [\lambda^O(\mathbf{u}) + \lambda^I(\mathbf{u})] - \lambda_j^I(u). \tag{3.7}$$

Conversely, the extrinsic LLR of the symbol $\mathbf{u}$ is obtained from the extrinsic LLR's of its component bits u_j as

$$\lambda(\mathbf{u}) = \sum_{j=1}^{m} \lambda_j u_j. \tag{3.8}$$

B. M-BCJR Algorithm

The M-BCJR algorithm [7] is a reduced-complexity variant of the BCJR algorithm and is based on the M-algorithm, a reduced-search trellis decoder. The reduction in complexity is achieved by retaining only the M-best paths in the forward recursion at each time instant. In the calculation of α_k through forward recursion on α_{k-1}, only the M largest components are used; the rest of them are set to an LLR of $-\infty$ and the corresponding states are thus declared dead. The backward recursion is executed only over those states that survived during the forward recursion. In Figure 6, we show an example of M-BCJR computation pattern for $M = 2$.

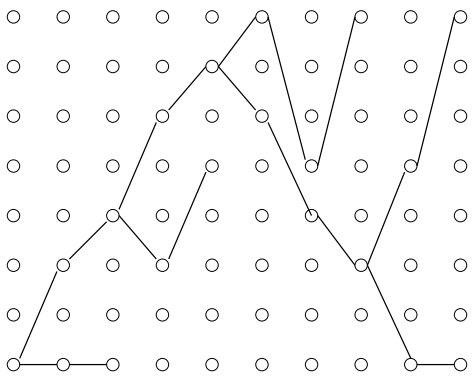

Fig. 6. Idealized computation pattern in the M-BCJR algorithm on an 8-state trellis. A line connecting two nodes indicates that the left node was used to compute the right node and that right node survived the reduction process

C. Theory of EXIT Charts

EXIT charts were originally proposed by Stephan ten Brink as a tool to better understand the convergence behavior of iterative decoding schemes [8]. The exchange of extrinsic information between constituent SISO modules is visualized as a decoding trajectory in the EXIT chart. This facilitates the prediction of convergence *thresholds* with reasonable accuracy besides helping to understand the influence of code memory, different constituent codes and precoders on the convergence behavior of concatenated codes. To plot an EXIT chart for an iterative decoder, we need to obtain the transfer characteristics of its constituent decoders.

1. Transfer Characteristics of Constituent Decoders

The basic purpose of EXIT charts is to predict the behavior of an iterative decoder by looking solely at the input/output relations of its constituent decoders. They are based on the following assumptions.

1. For large interleavers, the *a priori* values A remain fairly uncorrelated from their corresponding channel observations Z over many iterations.

2. The probability density function (pdf) of the output extrinsic LLR's E approaches Gaussian-like distributions as iterations progress [21].

The first observation can be explained by the fact that in the BCJR algorithm, the extrinsic output E_k at time instant k is not influenced by the channel observations Z_k or the *a priori* value A_k [5]. A large interleaver further contributes to reduce correlations. The possible reasons for the second observation are (a)the use of a Gaussian channel model, and (b)that sums over many values are involved in the calculation of E which typically leads to Gaussian-like distributions.

Before proceeding further to describe how to obtain the transfer characteristics of a constituent decoder, a review of some simple statistical properties of LLR values will be in order. Consider a SISO module with the channel observations Z and *a priori* LLR's A as its input and the extrinsic LLR's E as its output. For the received signal from the AWGN channel

$$z = x + n \tag{3.9}$$

the conditional pdf writes as

$$p(z|X = x) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left\{-\frac{1}{2\sigma_n^2}(z - x)^2\right\}. \tag{3.10}$$

where the binary random variable X with alphabet $\chi = \{-1, +1\}$ denotes the transmitted bits. The corresponding LLR's are calculated as

$$Z = \ln\left(\frac{p(z|x = +1)}{p(z|x = -1)}\right) \tag{3.11}$$

which simplifies to

$$Z = \frac{2}{\sigma_n^2} \cdot z = \frac{2}{\sigma_n^2} \cdot (x + n). \tag{3.12}$$

The variable n is Gaussian distributed with mean zero and variance $\sigma_n^2 = N_o/2$ (double-sided noise power spectral density). Equation (3.12) may be rewritten as

$$Z = \mu_Z \cdot x + n_Z \tag{3.13}$$

with

$$\mu_Z = \frac{2}{\sigma_n^2} \tag{3.14}$$

and n_Z being Gaussian distributed with mean zero and variance

$$\sigma_Z^2 = \frac{4}{\sigma_n^2}. \tag{3.15}$$

Thus, the mean and variance of Z are connected by

$$\mu_Z = \frac{\sigma_Z^2}{2}. \tag{3.16}$$

Based on the two observations made above, the *a priori* input A to the constituent decoder can be modeled by an independent Gaussian random variable n_A with variance σ_A^2 and mean zero in conjunction with the known transmitted information bits x.

$$A = \mu_A \cdot x + n_A. \tag{3.17}$$

Since A is supposed to be a Gaussian distributed LLR value as in the case of equations (3.13), (3.16), the mean value μ_A must fulfil

$$\mu_A = \frac{\sigma_A^2}{2}. \tag{3.18}$$

With (3.18), the conditional pdf belonging to A writes as

$$p_A(\xi|X = x) = \frac{1}{\sqrt{2\pi}\sigma_A} \exp\left\{-\frac{1}{2\sigma_A^2}(\xi - (\sigma_A^2/2) \cdot x)^2\right\}. \tag{3.19}$$

The equation (3.19) can also be derived using the consistency condition [22] for the LLR's distribution, $p_A(\xi|X = x) = p_A(-\xi|X = x)e^{x\cdot\xi}$.

Let $I_A = I(A; X)$ be the mutual information between the *a priori* values A and

the information bits X.

$$I_A = \frac{1}{2} \cdot \sum_{x=-1,+1} \int_{-\infty}^{+\infty} p_A(\xi|X=x) \times \log_2 \left\{ \frac{2 \cdot p_A(\xi|X=x)}{p_A(\xi|X=-1) + p_A(\xi|X=+1)} \right\} d\xi \tag{3.20}$$

$$0 \leq I_A \leq 1. \tag{3.21}$$

Similarly, $I_E = I(E;X)$ is defined as the mutual information between the output extrinsic LLR's E and the information bits X.

$$I_E = \frac{1}{2} \cdot \sum_{x=-1,+1} \int_{-\infty}^{+\infty} p_E(\xi|X=x) \times \log_2 \left\{ \frac{2 \cdot p_E(\xi|X=x)}{p_E(\xi|X=-1) + p_E(\xi|X=+1)} \right\} d\xi \tag{3.22}$$

$$0 \leq I_E \leq 1. \tag{3.23}$$

Viewing I_E as a function of I_A for a fixed E_b/N_o, the extrinsic information transfer characteristics are defined as

$$I_E = T(I_A). \tag{3.24}$$

2. Utility of Extrinsic Information Transfer Characteristics

To study the convergence behavior of an iterative decoder, the transfer characteristics of both the constituent decoders are plotted into a single chart. However, the characteristics of the second decoder (outer decoder in the case of SCCC's) are plotted with the axes swapped. The decoding process may be visualized as a trajectory in the EXIT chart. Here, we illustrate the utility of an EXIT chart for an SCCC. The procedure may be easily adapted to the case of PCCC's.

Let m be the iteration number. The signal-to-noise ratio E_b/N_o is assumed to be fixed. The subscripts i and o refer to the inner and outer decoders respectively. At

$m = 0$, the decoding starts at the origin with zero *a priori* knowledge $I_{A_i,0} = 0$. At iteration m, the extrinsic output of the inner decoder $I_{E_i,m} = T_i(I_{A_i,m})$. $I_{E_i,m}$ is passed as *a priori* information to the outer decoder and hence $I_{A_o,m} = I_{E_i,m}$. Interleaving does not change mutual information. The extrinsic output of the outer decoder is $I_{E_o,m} = T_o(I_{A_o,m})$, which is fed back to the inner decoder as *a priori* knowledge $I_{A_i,m+1} = I_{E_o,m}$ for the next iteration.

The decoding process evolves as long as $I_{E_o,m+1} > I_{E_o,m}$. With $I_{E_o,m+1} = T_o(T_i(I_{E_o,m}))$, this can be formulated as $T_i(I_{E_o,m}) > T_o^{-1}(I_{E_o,m})$. The decoding process willencounter a fixed point if $I_{E_o,m+1} = I_{E_o,m}$, or equivalently, $T_i(I_{E_o,m}) = T_o^{-1}(I_{E_o,m})$, which corresponds to an intersection of the two characteristics in the EXIT chart.

The behavior of the decoding trajectory in the EXIT chart gives rise to three regions of interest in the BER chart as discussed below.

1. *Pinch-off region*: The region of low E_b/N_o with imperceptible BER improvement with iterations. This happens when the decoder characteristics intersect at low mutual information and thus the decoding process encounters a fixed point.

2. *Bottle-neck region*: The turbo-cliff region where the decoding trajectory just manages to sneak through a narrow tunnel between the decoder characteristics; convergence toward low BER is slow but possible because the decoder characteristics no longer intersect. The lowest E_b/N_o at which this occurs is called as the *threshold* E_b/N_o.

3. *Wide-open region*: This is the region of fast convergence where the gap between the transfer characteristics of the component decoders is very wide. At very high E_b/N_o's, an error floor may surface.

As iterative decoding is block-oriented, we can further distinguish between *snapshot trajectory* and *averaged trajectory*. For the *snapshot trajectory*, the measurements of the extrinsic output pdf $p_{E,m}$ at iteration m stem from the decoding of a single block. For *averaged trajectory*, the extrinsic output pdf $p_{E,m}$ is obtained by averaging over a number of N_B blocks by applying

$$\overline{p_{E,m}(\xi|X=\pm 1)} = \frac{1}{N_B} \cdot \sum_{b=1}^{N_B} p_{E,n,b}(\xi|X=\pm 1) \tag{3.25}$$

to trace an average behavior of the decoding convergence in the EXIT chart.

The importance of EXIT charts lies in the fact that they help us analyze the decoding behavior of an iterative decoder with sufficient accuracy without the necessity for resource-intensive BER simulations. It should be noted that the actual decoding trajectory may deviate from the expected path in the EXIT chart. This is attributed to the increasing correlation of the extrinsic information with the information bits as the iterations progress. The observed deviations are smaller when larger interleavers are used. The EXIT charts may also be used to estimate the achievable channel capacity with iterative decoding [23]. This is due to the property that for any outer code, if the extrinsic information on the coded bits is assumed to be coming from an erasure channel, the area under its transfer characteristic is the rate of the outer code. Although this property holds only when the extrinsic channel is an erasure channel, we observe that this is almost true in most practical situations and hence we will assume that the above property holds good for IED as well. This implies that the achievable information rate for a given equalizer in IED is the area under the transfer characteristic of the equalizer which we shall loosely refer to as the achievable channel capacity.

CHAPTER IV

PERFORMANCE ANALYSIS OF M-BCJR EQUALIZER

The performance of the M-BCJR equalizer is studied and contrasted with that of the BCJR equalizer on a variety of ISI channels (precoded and non-precoded) with the help of BER and FER simulations and EXIT charts [8].

The ISI channels are modeled as convolutional codes (DTTF's) as discussed in Chapter II. In all our simulations, the channel is assumed to be static and its coefficients $f_m,\ m = 0, 1, \ldots, L-1$ where L is the length of the channel impulse response, are perfectly known. Each of the channel coefficients has a power equal to

$$P_m = |f_m^2|, \quad m = 0, 1, \ldots, L-1$$

and is normalized such that the total power

$$\sum_{m=0}^{L-1} P_m = 1.$$

In this chapter and the remainder of the thesis, we shall represent an ISI channel by its coefficients $[f_0, f_1, \ldots, f_{L-1}]$.

We also investigate the performance of the M-BCJR equalizer on precoded channels because precoding improves the asymptotic performance of SCC's [15]. Precoding is achieved by appropriately processing the interleaved bits stream prior to passing it through the DTTF.

A. Performance of the M-BCJR Equalizer

In Figure 7, we show the BER and FER performance of IED for the unprecoded $[\sqrt{0.45}, \sqrt{0.25}, \sqrt{0.15}, \sqrt{0.10}, \sqrt{0.05}]$ channel with $[1, 3]_8$ as the outer code using 10 iterations and N=2048. The loss in performance of the M-BCJR equalizer as com-

pared to the BCJR equalizer is unexpectedly small for non-precoded ISI channels. The BER and FER curves of the M=8 BCJR algorithm are almost overlapping with those of the full BCJR which operates on the whole 16-state trellis. When M=4 is used, the loss in performance is only 0.05 dB at a BER of 10^{-5}. For M=3, the loss is 0.25 dB at a BER of 10^{-4}. For M=2, the IED algorithm fails to evolve and does not provide any improvement in performance with iterations.

As can be seen from these results for the above channel, we may use the M=4 BCJR equalizer with virtually no performance degradation or the M=3 BCJR equalizer with a very small loss in performance. This is an interesting result and suggests that the complexity of the BCJR equalizer can be reduced considerably without sacrificing its performance.

In the remainder of this section, we present a few more simulation results to show that the M-BCJR equalizer delivers similar performance on a majority of ISI channels, if not all of them. In fact, we present stronger simulation results in Section B based on EXIT charts which demonstrate that the M-BCJR equalizer suffers negligible losses on practically every ISI channel even for very small values of M.

In Figure 8, we present the BER and FER simulation results over 6 iterations for the unprecoded $[0.5, 0.5, 0.5, 0.5]$ channel using the full BCJR and M=3 BCJR equalizers. An information block length of N=2048 was used. The full BCJR equalizer operates on the whole trellis consisting of 8 states at each time instant. From the plots, we observe that the performance of the M=3 BCJR equalizer is almost indistinguishable from that of the full BCJR equalizer in the region of BER=10^{-5} at reasonably high E_b/N_o. The performance of the M=3 BCJR equalizer is relatively worse at low E_b/N_o.

The performance of the turbo equalizer saturates at a BER of 10^{-5} and does not improve significantly even at very high E_b/N_o. Such an early error floor is typical of

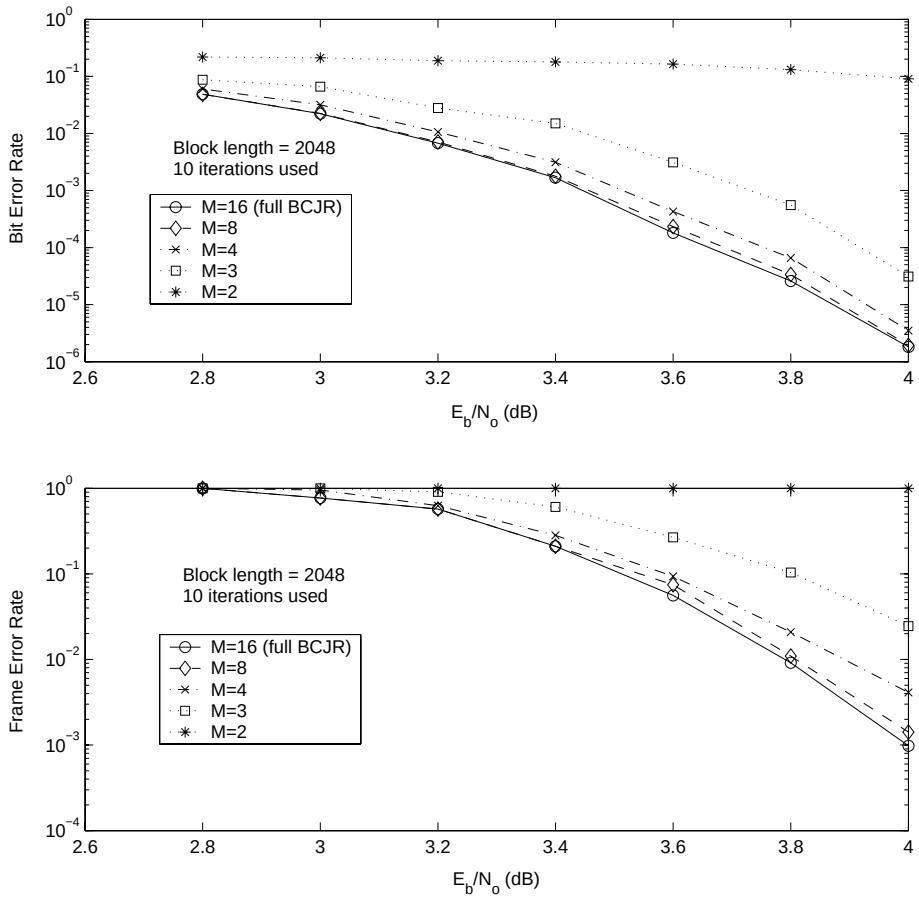

Fig. 7. BER & FER: Non-precoded $[\sqrt{0.45}, \sqrt{0.25}, \sqrt{0.15}, \sqrt{0.10}, \sqrt{0.05}], [1, 3]_8$

SCC's in which the inner code is non-recursive [15]. It can be avoided by precoding the channel.

The BER and FER simulation results for the precoded $[0.5, 0.5, 0.5, 0.5]$ channel

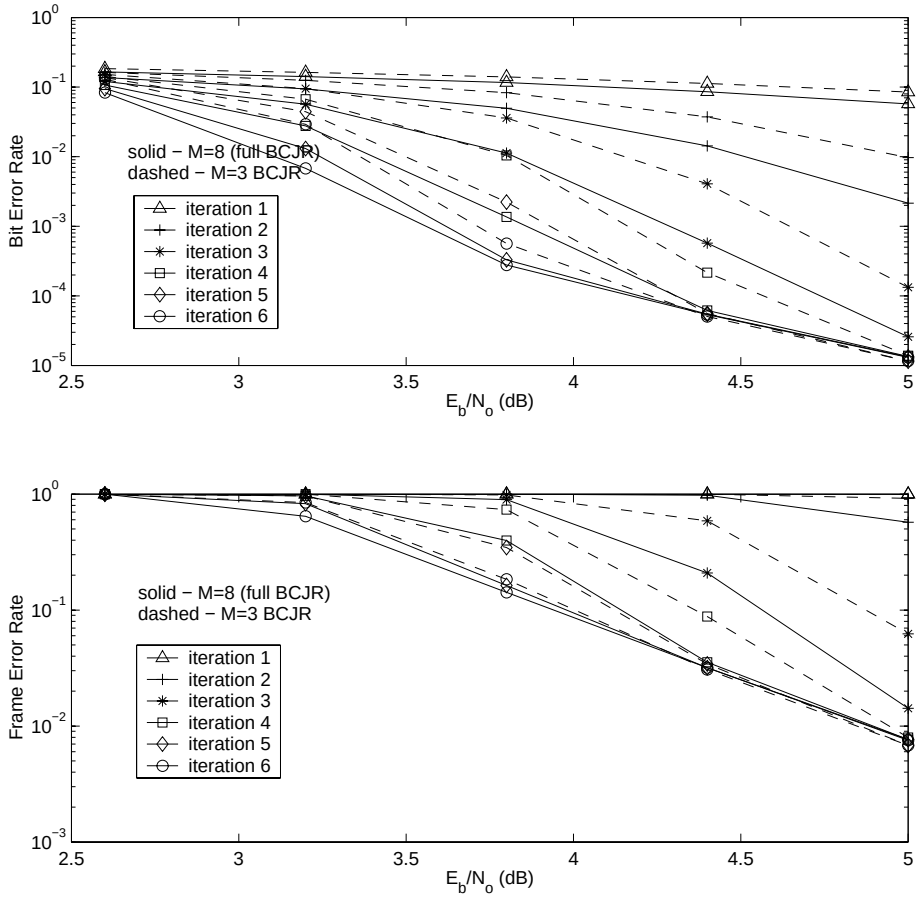

Fig. 8. Simulation results: Non-precoded $[0.5, 0.5, 0.5, 0.5]$, $[1, \frac{23}{35}]_8$

using the full BCJR, M=5 and M=3 BCJR equalizers are plotted in Figure 9. In comparison with the non-precoded channel, the M-BCJR equalizer suffers significant losses in the precoded case. The performance of the M=5 BCJR equalizer is approximately 0.25 dB worse than the BCJR algorithm at a BER of 10^{-4}. However, this difference diminishes as we progress toward smaller BER's. For M=3 BCJR, the IED hardly yields any improvement in the BER performance with increasing number

of iterations. This is in stark contrast with its performance on the non-precoded channel. It is also interesting to note that although the asymptotic performance on precoded channels is better at high E_b/N_o, the non-precoded channels offer better performance during the first few iterations. This behavior is a result of the fact that the initial reliability of a precoded channel is smaller than that of a non-precoded channel [24]. However, as the iterations progress, the precoded channel outperforms the non-precoded channel. Also, there are no signs of an error floor at a BER of 10^{-5} and thus the performance may improve significantly as E_b/N_o increases.

In Figures 10 and 11, we plotted the BER and FER simulation results for

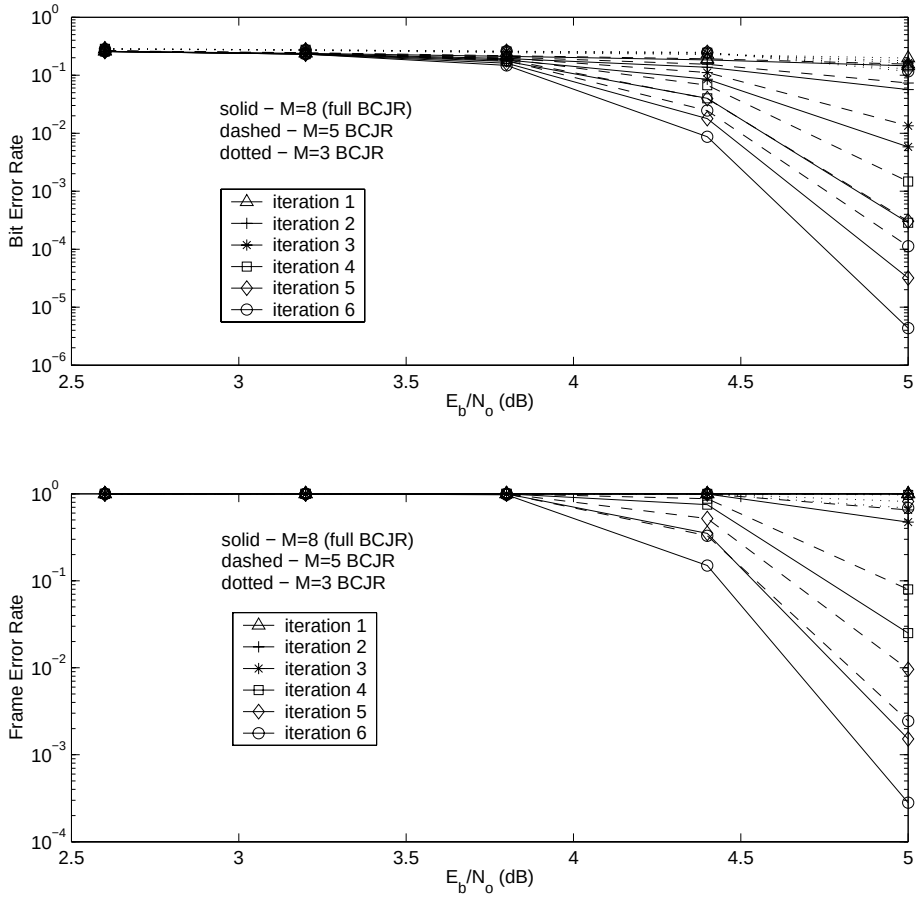

Fig. 9. Simulation results: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

the non-precoded and precoded $[0.5, 0.5, 0.0, 0.5, 0.5]$ channels respectively. A block length of N=2048 and 6 iterations were used. For the non-precoded channel, the performance graphs at the end of 6 iterations for the M=3 BCJR and the BCJR equalizers are almost overlapping as in the case of the non-precoded $[0.5, 0.5, 0.5, 0.5]$ channel. For the precoded channel, the performance loss of the M=8 BCJR equalizer is very small. For M=3 BCJR, the IED fails to improve the performance with any number of iterations. All remarks made earlier with regards to the equalizer performance on the precoded and non-precoded $[0.5, 0.5, 0.5, 0.5]$ channel are applicable to this channel as well.

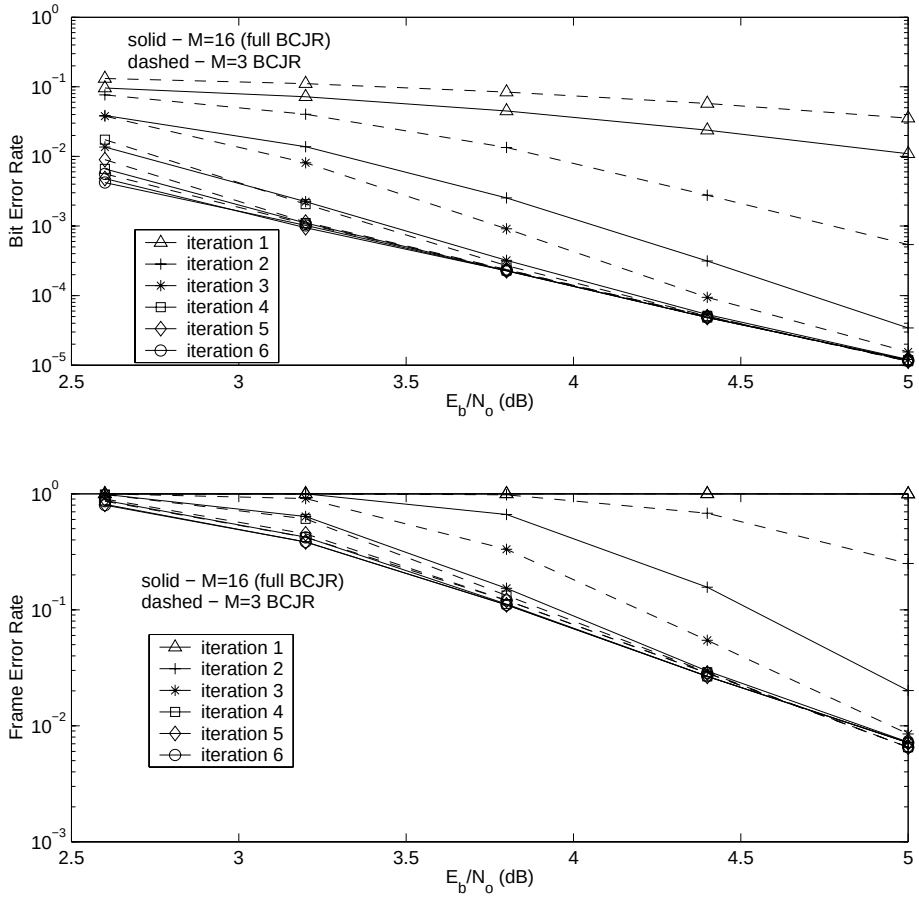

Fig. 10. Simulation results: Non-precoded $[0.5, 0.5, 0.0, 0.5, 0.5]$, $[1, \frac{23}{35}]_8$

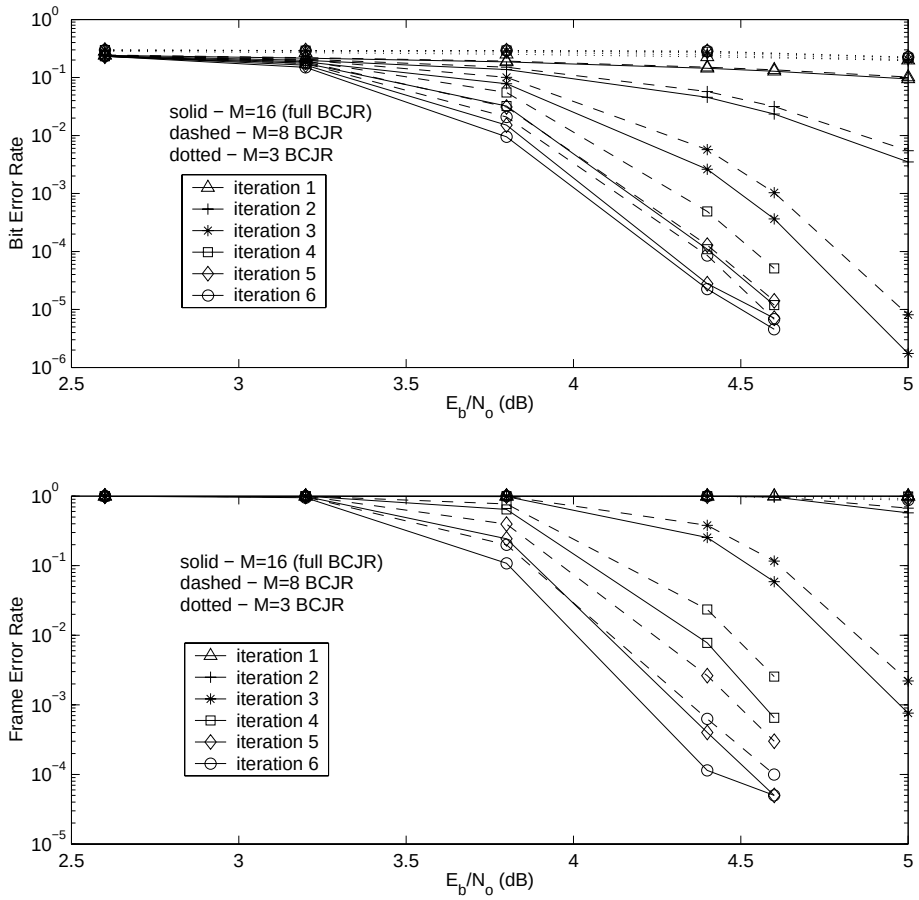

Fig. 11. Simulation results: 1⊕D precoded $[0.5, 0.5, 0.0, 0.5, 0.5], [1, \frac{23}{35}]_8$

B. Analysis of the Performance of the M-BCJR Equalizer

In this section, we use EXIT charts to understand and analyze the convergence behavior of the M-BCJR equalizer on several precoded and non-precoded ISI channels.

1. Explanation of Simulation Results

In Figure 12, we present five EXIT charts for the unprecoded $[0.5, 0.5, 0.5, 0.5]$ channel with $[1, 23/35]_8$ as the outer code at different values of E_b/N_o. They are obtained by averaging the pdf's, p_A and p_E of each decoder over five blocks each corresponding to an information length of $250,000$ bits. The transfer characteristics of the outer decoder are drawn with the axes swapped. The gaps between the transfer characteristics of different equalizers and the decoder are good indicators of the relative speed of convergence of turbo equalization. The more the gap, the faster the convergence.

Let us first consider the full BCJR equalizer. The EXIT chart at $E_b/N_o = 2.6$ dB shows that the transfer characteristics of the equalizer and the decoder intersect at a low $I_E(=0.65)$. Thus the IED is operating in the *pinch-off region* and does not provide significant BER improvement. As the E_b/N_o increases, the point of intersection moves towards higher I_E subsequently providing improved BER performance. At $E_b/N_o = 5.0$ dB, the decoder characteristics intersect at $I_E = 0.82$ which is reasonably high and thus gives acceptable BER performance. The wide gap between the transfer characteristics of the component decoders ensures fast convergence and the IED may be regarded as operating in the *wide-open region*. Also, as the EXIT charts show, the asymptotic extrinsic output mutual information I_E when the *a priori* mutual information $I_A = 1.0$ is always less than unity for non-precoded channels. This explains the saturation of IED on non-precoded channels as seen in Figures 8 and 10.

The EXIT charts also depict the performance of the M=3 BCJR equalizer. The

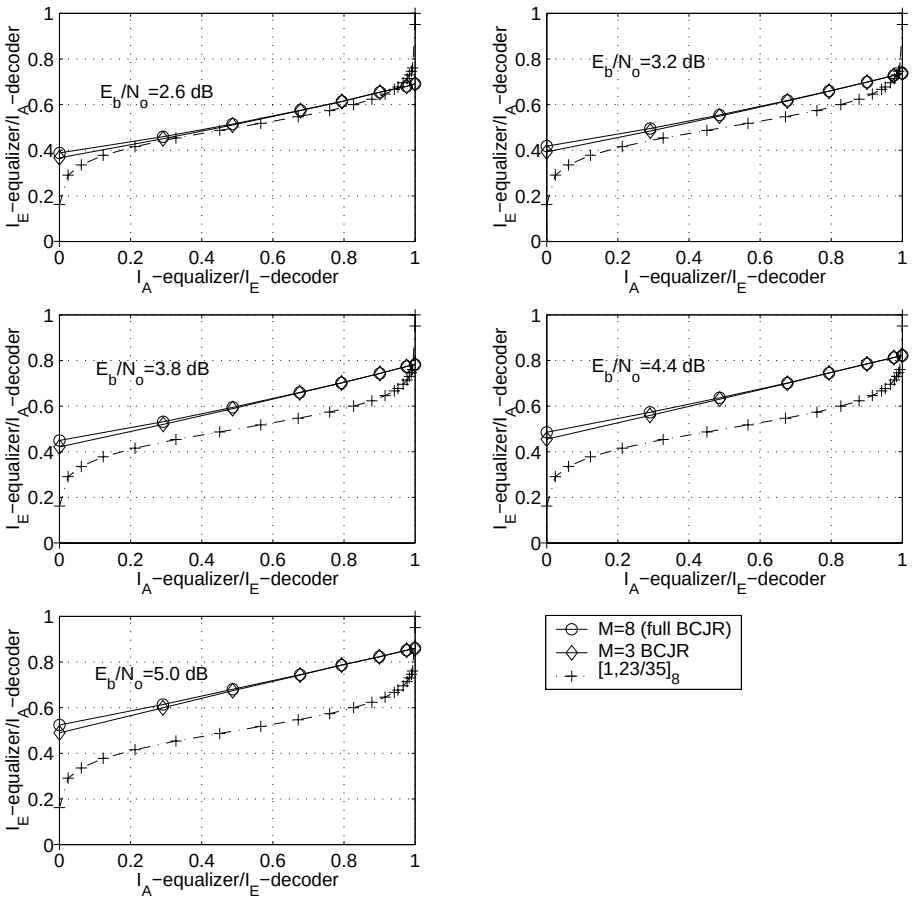

Fig. 12. EXIT charts: Non-precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

transfer characteristics of the M=3 equalizer are very close to the full BCJR equalizer at all E_b/N_o. In fact, at high values of I_A they merge. Based on this observation, we expect the performance of the M=3 equalizer to be slightly degraded during the initial iterations but nearly the same as the BCJR equalizer after sufficient number of iterations. Since the gap between the characteristics of the M=3 equalizer and the decoder is slightly smaller, the speed of convergence will be relatively slower. An examination of the BER charts in Figure 8 reveals that the actual performance is in slight disagreement with our expectations. The gap between the performances of the BCJR and the M=3 equalizers closes after six iterations at high E_b/N_o but is significantly wide at low E_b/N_o. We shall briefly explain this behavior here. A detailed analysis and explanation is the object of Chapter V. The extrinsic outputs of the M-BCJR equalizer are not true LLR's. However, as more and more *a apriori* information becomes available, they approach true LLR's. An EXIT chart is plotted with the assumption that the extrinsic outputs are used in an optimal fashion and therefore their falsity does not degrade the asymptotic performance. Hence the performance of an M-BCJR equalizer as predicted from its transfer characteristics will necessarily be optimistic. Thus at low E_b/N_o, the true gap between the decoder characteristics would be narrower and the observed BER performance is significantly degraded perhaps due to an early intersection. On the other hand, at high E_b/N_o, the true gap would still be sufficiently wide thus ensuring convergence, although at a slower rate.

The achievable channel capacities for the BCJR and M=3 equalizers at different E_b/N_o are listed in Table I. The M=3 equalizer provides almost the same channel capacity as the BCJR equalizer.

The EXIT charts for the unprecoded $[0.5, 0.5, 0.0, 0.5, 0.5]$ channel with $[1, 23/35]_8$ as the outer convolutional code are plotted in Figure 13. They were obtained by av-

Table I. Channel capacities: Non-precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

E_b/N_o (dB)	Equalizer	
	M=8 (full BCJR)	M=3 BCJR
2.6	0.5256	0.5191
3.2	0.5650	0.5574
3.8	0.6044	0.5960
4.4	0.6454	0.6360
5.0	0.6864	0.6763

eraging over 5 blocks of length N=250,000 each. Again, we observe in Figure 13 that the error performances of the full BCJR and M=3 equalizers are almost coincident at high E_b/N_o after 6 iterations. The performance of M=3 equalizer is only slightly degraded at low E_b/N_o and approaches the full BCJR performance at high E_b/N_o. However, the rate of convergence is relatively slow due to the narrower gap between the component decoder characteristics. Also, the asymptotic I_E is always less than unity as noted earlier which results in early saturation of BER in IED. These results can be explained by applying the same arguments as used above for the unprecoded $[0.5, 0.5, 0.5, 0.5]$ channel. The estimated channel capacities at different E_b/N_o are listed in Table II. It shows that the $M = 3$ BCJR equalizer offers almost the same capacity as the BCJR equalizer for this channel also.

Next, the EXIT charts of the precoded $[0.5, 0.5, 0.5, 0.5]$ and $[0.5, 0.5, 0.0, 0.5, 0.5]$ channels are presented in Figures 14 and 15 respectively. These were also obtained by averaging over 5 blocks of length of N=250,000 each. The EXIT charts for the precoded $[0.5, 0.5, 0.5, 0.5]$ channel show that the transfer characteristics of the BCJR equalizer and the decoder intersect at $E_b/N_o < 3.8$ dB. At $E_b/N_o = 3.9$ dB, the BCJR turbo equalizer can just manage to sneak through the narrow tunnel between the decoder characteristics thus providing good BER performance albeit at a slow convergence rate. In fact, as we shall see in Chapter V, the actual performance of the

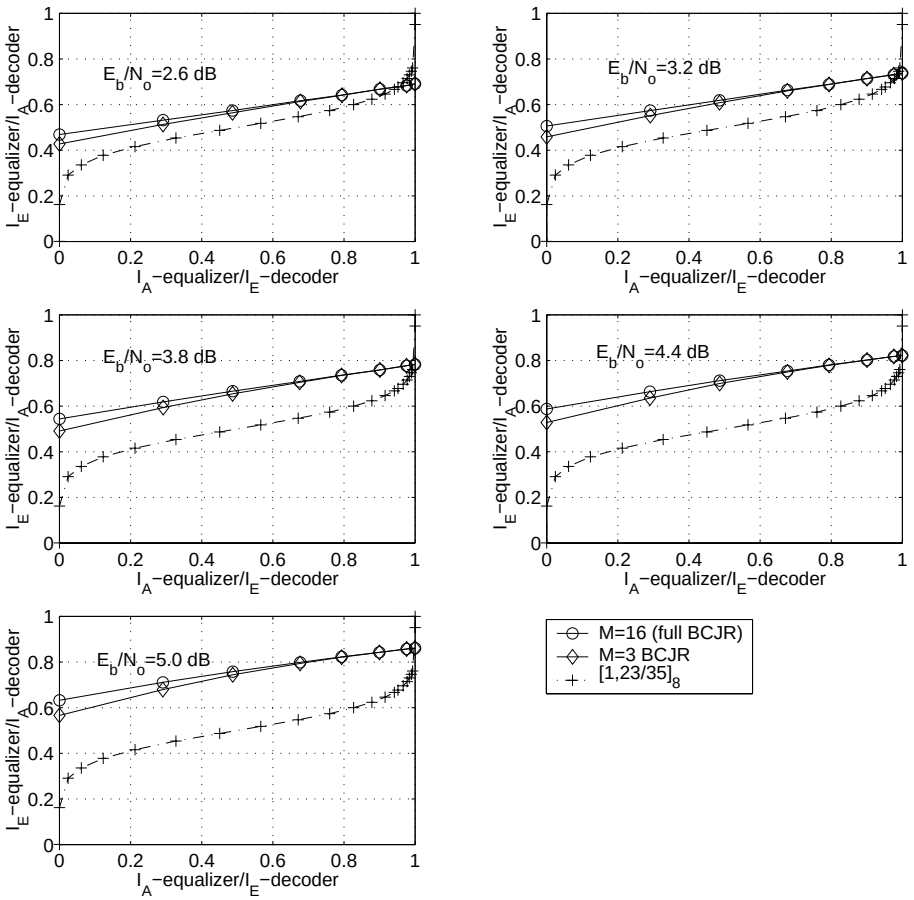

Fig. 13. EXIT charts: Non-precoded $[0.5, 0.5, 0.0, 0.5, 0.5], [1, \frac{23}{35}]_8$

Table II. Channel capacities: Non-precoded $[0.5, 0.5, 0.0, 0.5, 0.5], [1, \frac{23}{35}]_8$

E_b/N_o (dB)	Equalizer	
	M=16 (full BCJR)	M=3 BCJR
2.6	0.5772	0.5642
3.2	0.6213	0.6063
3.8	0.6658	0.6488
4.4	0.7104	0.6915
5.0	0.7550	0.7340

BCJR equalizer exhibits convergence at the expected threshold of $E_b/N_o = 3.9$ dB. At $E_b/N_o \geq 3.9$ dB, the BCJR turbo equalizer provides rapidly improving BER performance as E_b/N_o increases.

The simulation results in Figure 9 however, show that the performance just begins to improve at $E_b/N_o = 3.8$ dB and provides acceptable error performance of BER$\leq 10^{-5}$ only at $E_b/N_o \geq 5.0$ dB approximately. This behavior is in contradiction with our expected threshold of $E_b/N_o = 3.9$ dB but can be easily explained by noting that the EXIT charts only portray the asymptotically best achievable performance. The discrepancy in actual performance as seen in Figure 9 is attributed to the small information block length of N=2048 (or equivalently the small interleaver size) and the small number of iterations (only 6) used for obtaining these plots. At $E_b/N_o = 3.8$ dB, the performance just begins to improve but does not converge due to the small block length. At $3.8 < E_b/N_o < 5.0$, there is only a gradual improvement in performance. The small block length is still a significant limiting factor in this region but the performance is also affected by the small number of iterations used. In other words, the BER may be significantly improved simply by using more iterations.

The M=5 BCJR equalizer intersects the transfer characteristic of $[1, 23/35]_8$ outer decoder at $E_b/N_o < 4.0$ dB as may be deduced from the EXIT charts. Accordingly, its performance just begins to improve at $E_b/N_o = 4.0$ dB and gradually

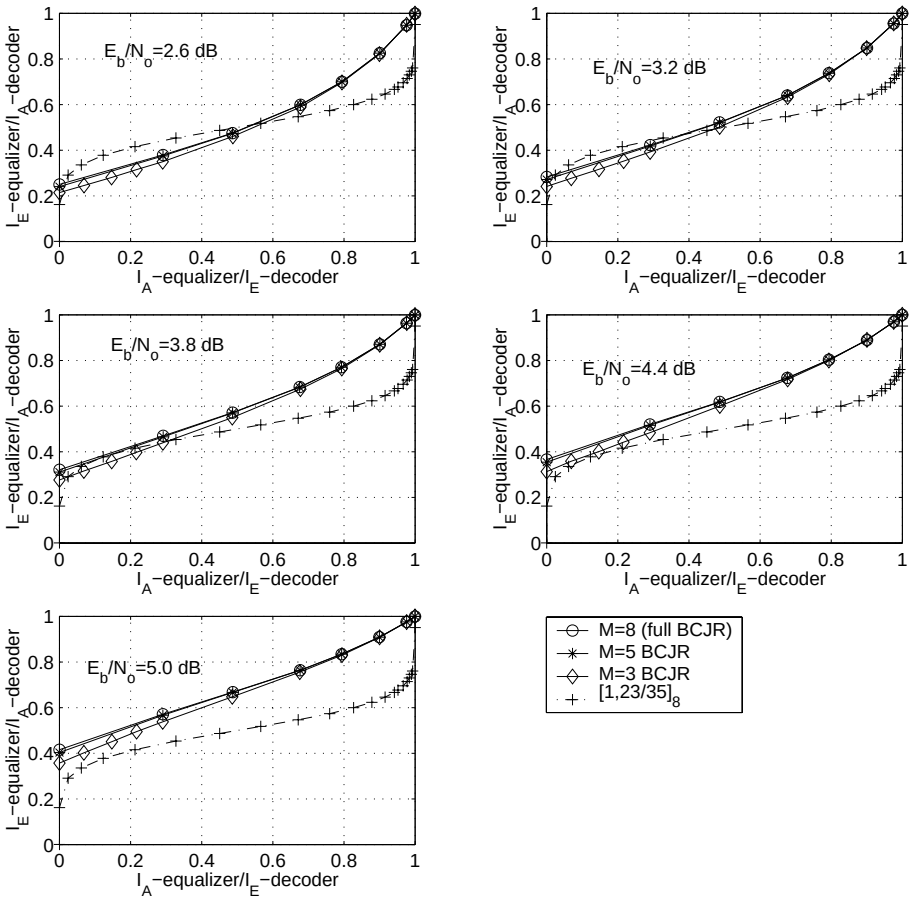

Fig. 14. EXIT charts: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

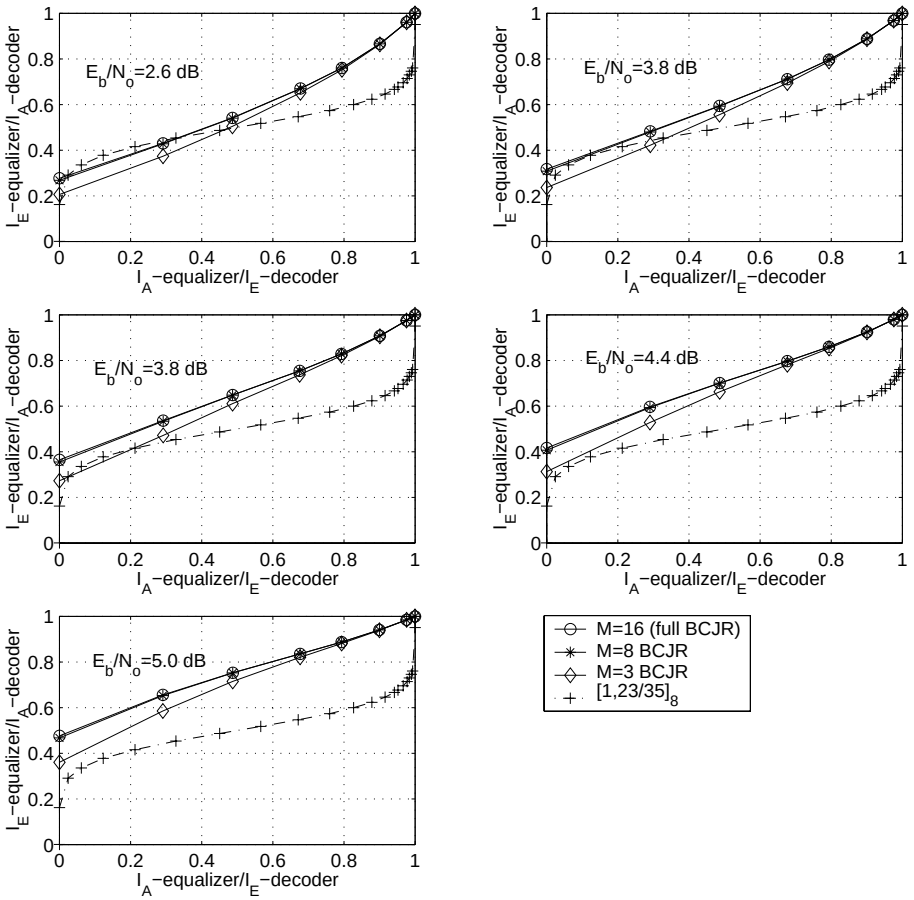

Fig. 15. EXIT charts: 1⊕D precoded $[0.5, 0.5, 0.0, 0.5, 0.5], [1, \frac{23}{35}]_8$

improves as E_b/N_o increases. The convergence rate is slow as compared with the BCJR equalizer because of the narrower gap between the characteristics of the component decoders.

As already noted, the extrinsic outputs of M-BCJR equalizers are not true LLR's and therefore the performance predicted from their EXIT charts would seem optimistic. Hence the simulated performance of M-BCJR equalizers may actually be more degraded than expected. This phenomenon is mirrored in the M=3 equalizer performance. Based on EXIT charts, we expect a threshold of $E_b/N_o = 4.4$ dB for the M=3 equalizer. However, as the simulation results in Figure 9 show, it does not provide any improvement in BER at least for $E_b/N_o \leq 5.0$ dB. In fact, we shall see in Chapter V that it evolves at $E_b/N_o \approx 5.1$ dB using a large information block length (N=100,000) and by decoding over 15 iterations.

In Figure 15, we plot the EXIT charts of the precoded $[0.5, 0.5, 0.0, 0.5, 0.5]$ channel for the BCJR, M=8 and M=3 equalizers. The BER performance of different turbo equalizers on this channel shown in Figure 11 may be explained from the EXIT charts by using similar arguments as for the precoded $[0.5, 0.5, 0.5, 0.5]$ channel. The transfer characteristics of the BCJR and the M=8 equalizers are very close in the region of low I_A and merge at high I_A. This explains why the performance of the M=8 equalizer approaches that of the BCJR equalizer after sufficient number of iterations at high E_b/N_o. On the other hand, though a threshold of 4.4 dB is expected for the M=3 equalizer, it actual performance is severely degraded and does not show any improvement in performance at least for $E_b/N_o \leq 5.0$ dB. Once again, this observation is justified by the fact that the extrinsic outputs of M-BCJR equalizers are not true LLR's and thus the EXIT charts provide optimistic predictions about their performance.

The channel capacity estimates for the precoded channels: $[0.5, 0.5, 0.5, 0.5]$ and

Table III. Channel capacities: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

E_b/N_o (dB)	Equalizer		
	M=8 (full BCJR)	M=5 BCJR	M=3 BCJR
2.6	0.5239	0.5212	0.5064
3.2	0.5615	0.5586	0.5425
3.8	0.6010	0.5979	0.5805
4.4	0.6409	0.6378	0.6199
5.0	0.6815	0.6783	0.6600

Table IV. Channel capacities: 1⊕D precoded $[0.5, 0.5, 0.0, 0.5, 0.5], [1, \frac{23}{35}]_8$

E_b/N_o (dB)	Equalizer		
	M=16 (full BCJR)	M=8 BCJR	M=3 BCJR
2.6	0.5752	0.5732	0.5401
3.2	0.6170	0.6149	0.5794
3.8	0.6600	0.6578	0.6199
4.4	0.7039	0.7018	0.6616
5.0	0.7476	0.7456	0.7033

$[0.5, 0.5, 0.0, 0.5, 0.5]$ at different E_b/N_o can be found in Tables III and IV respectively. These tables reiterate that the capacity loss of M-BCJR equalizers is negligibly small. A comparison of these tables with those of the corresponding non-precoded channels shows that precoding does not affect the achievable channel capacity.

2. Comparison of M-BCJR and MMSE Equalizers

In Figure 16, the EXIT chart of the exact MMSE LE adopted from [13] on the unprecoded $[0.227, 0.46, 0.688, 0.46, 0.227]$ channel is plotted for comparison with the BCJR and M-BCJR equalizers. The estimates of channel capacities for various equalizers are also listed on the figure. These estimates show that the MMSE equalizer suffers a substantial loss in channel capacity compared with the BCJR equalizer. On the other hand, the capacities of M=5 and M=3 BCJR equalizers are quite close to the BCJR equalizer's capacity. Therefore the loss in capacity suffered by using an M-BCJR

equalizer instead of the BCJR equalizer is very small.

In Figures 17 and 18, we plot the cdf's of achievable channel capacity loss

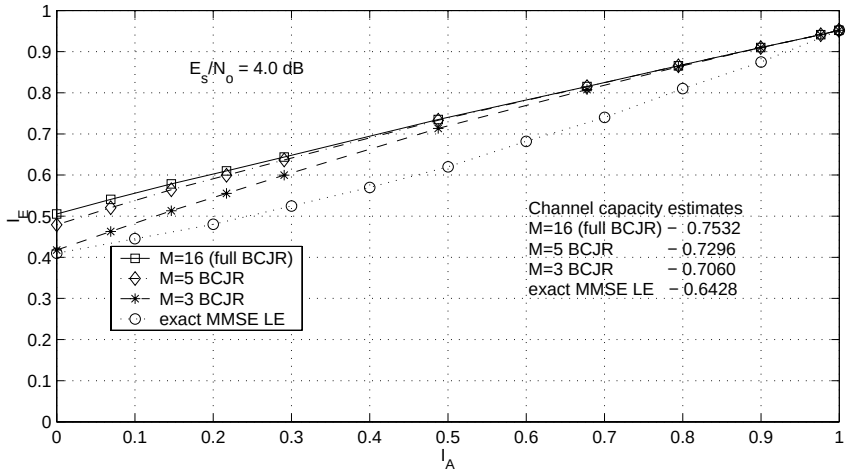

Fig. 16. Comparison of BCJR, M-BCJR and exact MMSE LE on non-precoded $[0.227, 0.46, 0.688, 0.46, 0.227]$

between BCJR and M=3 BCJR equalizers for 4-tap (L=4) and 5-tap (L=5) ISI channels respectively. The channel coefficients $f_m, m = 0, 1, \ldots, L-1$ where L is the length of the channel impulse response were randomly generated subject to the constraint that

$$\sum_{m=0}^{L-1} |f_m|^2 = 1.$$

For each random channel, the capacity losses at three E_s/N_o's of 1.4 dB, 2.0 dB and 2.6 dB are plotted in Figures 17 and 18. A data length of 250,000 was used for all simulations. The sample sizes for 4-tap and 5-tap channels are 5150 and 3050 respectively.

The loss in capacity for 4-tap channels was less than 2% for a majority of them. At worst, the loss was only 5% approximately. For 5-tap channels, a majority of

channels suffered a loss of less than 3% and the worst loss was 6% or less. These observations suggest that the performance of the M-BCJR equalizer is almost the same as the BCJR equalizer. Also, the loss in channel capacity is very small.

With a fixed outer decoder during IED, the shape of the MMSE transfer char-

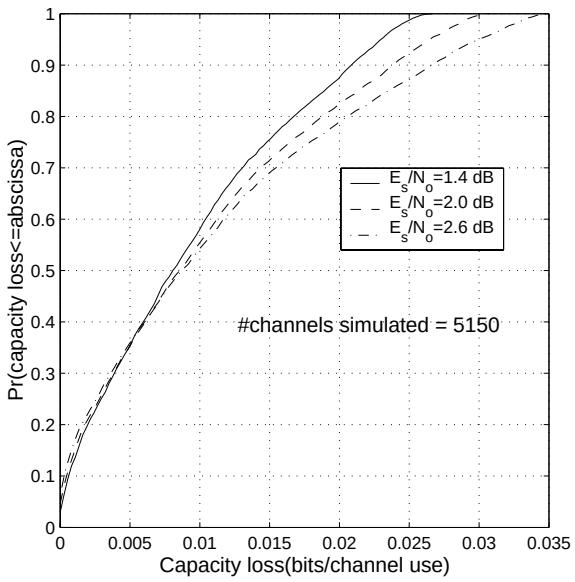

Fig. 17. Capacity loss cdf: 4-tap channels

acteristic suggests that its *threshold* E_b/N_o may be higher or the rate of convergence may be much slower or both as compared with an M-BCJR equalizer. Hence the advantage of using an M-BCJR equalizer over an MMSE equalizer is two-fold. It should however be remembered that the complexity of an MMSE equalizer is several times smaller.

C. M-BCJR Algorithm for Convolutional Codes

In Section A, we saw that the performance degradation of the M-BCJR equalizer in relation to the BCJR equalizer is negligibly small even for small values of M.

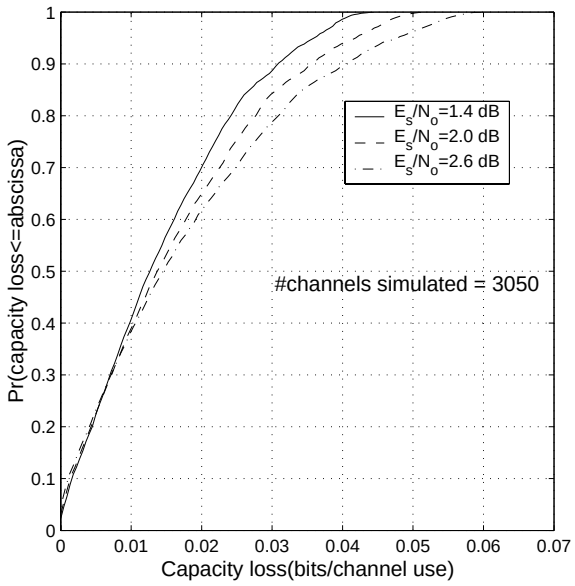

Fig. 18. Capacity loss cdf: 5-tap channels

These observations were also supported by the EXIT charts presented in Section B. Figures 19 and 20 present the EXIT charts for the BCJR and M-BCJR algorithms on some traditional convolutional codes. These plots show that the performance of the M-algorithm as compared with the BCJR algorithm is severely degraded. Thus, we see that the M-algorithm behaves differently in the cases of convolutional codes and DTTF models. Here, we try to explain this anomaly with the help of some empirical arguments.

As explained in Chapter III, the M-BCJR algorithm differs from the original BCJR only in the calculation of $\alpha_k(s)$ and $\beta_k(s)$ during the forward and backward recursions. They are respectively computed using equations (3.4) and (3.5). In the original BCJR algorithm, the number of terms in the numerator (or denominator) of (3.4) and (3.5) is a constant. But, in case of the M-BCJR algorithm, the number of active terms in the numerator (or denominator) of these equations varies depending on the survivor states at the preceding or the succeeding time instant. However, the

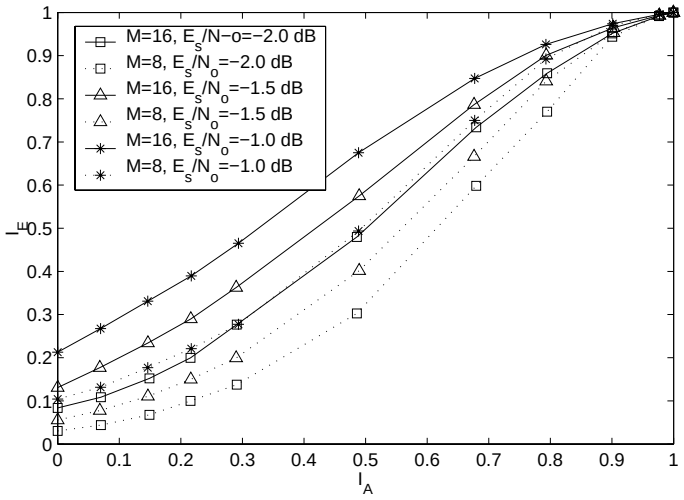

Fig. 19. EXIT charts: $[1, \frac{23}{35}]_8$

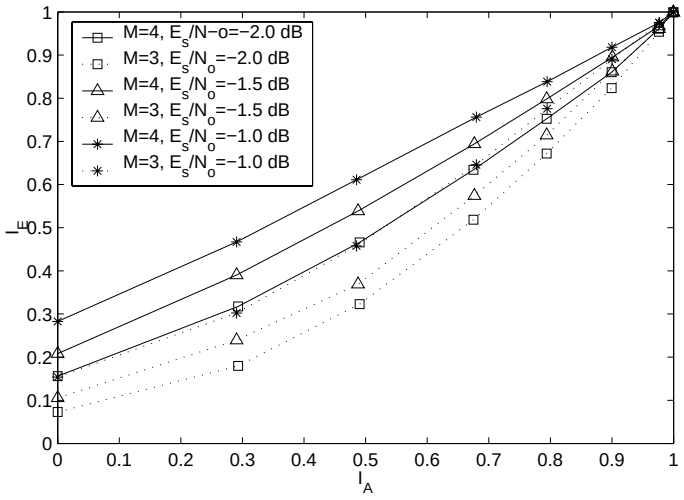

Fig. 20. EXIT charts: $[1, \frac{7}{5}]_8$

number of such terms is always less than the constant number of terms on which the BCJR operates. We note that $\overset{*}{\log}$ defined in equation (3.1) is the main operator used in both these recursive equations. The performance degradation of the M-algorithm therefore depends on the sensitivity of the $\overset{*}{\log}$ operator to its arguments.

The computation of the $\overset{*}{\log}$ operation may be greatly simplified as noted in equation (3.6) with only a small performance degradation. This tells us that the $\overset{*}{\log}$ operator may not be sensitive to some of its arguments if they are relatively small in value. Let a_1, a_2 be two values. Without loss of generality, assume that $a_1 > a_2$. Then,

$$\overset{*}{\log}(a_1, a_2) = \max(a_1, a_2) + \delta(a_1, a_2) \approx a_1 \tag{4.1}$$

if $(a_1 - a_2)$ is sufficiently large.

Based on the above arguments, we may conclude that the M-BCJR algorithm suffers more degradation in performance if the variance of the arguments entering the $\overset{*}{\log}$ operation in equations (3.4) and (3.5) is small. If the variance is large, the loss in performance will be less. To explain the contrasting behavior of M-BCJR algorithm on convolutional codes and ISI channels, we show that the variance of metrics entering the calculation of $\alpha_k(s)$ and $\beta_k(s)$ is much larger for ISI channels than for convolutional codes.

We first present some empirical arguments to show that the variance of metrics is larger for convolutional codes. Later, we present more reliable simulation results to support our explanation. Let $w_k, k = 1, 2, \ldots, K$ be the discrete-time received signal used to compute the channel observation LLR's $L_j(w_k), k = 1, 2, \ldots, K; j =$

$1, 2, \ldots, J$ using

$$L_j(w_k) = -\frac{1}{2\sigma_n^2} \cdot (w_k - s_j)^2 - \log \left\{ \sum_{j=1}^{J} \exp \left\{ -\frac{1}{2\sigma_n^2} \cdot (w_k - s_j)^2 \right\} \right\} \tag{4.2}$$

where $s_j, j = 1, 2, \ldots, J$ forms the output alphabet of the convolutional encoder or the ISI channel and σ_n^2 is the noise variance. $\alpha_k(s)$ and $\beta_k(s), k = 1, 2, \ldots, K; s = 1, 2, \ldots, M$ where M is the number of trellis states, are calculated by normalized recursive accumulation of $L_j(w_k)$'s over a number of time instants. Since at any time instant the number of $L_j(w_k)$'s entering the computation of any $\alpha_k(s)$ or $\beta_k(s)$ may be considered to be approximately the same, we infer that the variance of the metrics used in these computations varies directly in proportion with the variance of $L_j(w_k)$'s. Ignoring the constant in equation (4.2), the variance B_k of $L_j(w_k)$'s is given by

$$B_k = -\frac{1}{2J\sigma_n^2} \left\{ \sum_{j=1}^{J} (w_k - s_j)^2 \right\}. \tag{4.3}$$

Consider a typical convolutional encoder whose output alphabet is BPSK. Using equation (4.3), its variance B_k^{conv} at a particular time instant k computes as

$$B_k^{conv} = \frac{w_k^2}{\sigma_n^4}. \tag{4.4}$$

The output alphabet of an ISI channel typically has more than two symbols. For example, the output alphabet of $[0.5, 0.5, 0.5, 0.5]$ channel (precoded or non-precoded) is $\{-2, -1, 0, +1, +2\}$ and its variance B_k^{isi} computes as

$$B_k^{isi} = \frac{2w_k^2}{\sigma_n^4} + \frac{0.7}{\sigma_n^4}. \tag{4.5}$$

From equations (4.4) and (4.5), we infer that the variance of metrics for ISI channels is larger than for convolutional codes. In Tables V and VI, we present some simulation results to support our conclusion. They list the metric variances at

Table V. Metric variances in $\alpha_k(s)$ computation on some ISI channels

Channel	1⊕D precoded $[0.5, 0.5, 0.5, 0.5]$		Non-precoded $[0.5, 0.5, 0.5, 0.5]$		Non-precoded $[0.5, 0.5, 0.0, 0.5, 0.5]$	
$E_s/N_o(dB)$	2.6	3.2	2.6	3.2	-0.4	0.2
$\sigma_A = 0$	20.82	26.77	20.83	26.73	7.48	9.21
$\sigma_A = 4$	35.30	42.50	40.65	47.89	26.10	29.28
$\sigma_A = 10$	67.03	76.08	84.81	94.65	72.32	77.13
$\sigma_A = 15$	100.80	111.583	132.39	144.71	124.45	131.46
$\sigma_A = 25$	194.12	206.411	256.88	274.17	265.72	275.01

Table VI. Metric variances in $\alpha_k(s)$ computation on some convolutional codes

Channel	$[1, \frac{23}{35}]_8$		$[1, \frac{7}{5}]_8$	
$E_s/N_o(dB)$	-2.0	-1.5	-2.0	-1.5
$\sigma_A = 0$	0.31	0.41	0.34	0.46
$\sigma_A = 4$	1.73	2.15	1.89	2.26
$\sigma_A = 10$	7.59	9.04	6.48	7.48
$\sigma_A = 15$	15.50	17.57	12.41	13.91
$\sigma_A = 25$	37.80	41.19	29.04	31.98

various E_s/N_o for ISI channels and convolutional codes respectively. The *a priori* LLR's are simulated with a variance σ_A^2 and mean $\mu_A = \sigma_A^2/2$ in accordance with equation (3.16). As predicted, the variance of metrics used in the computation of $\alpha_k(s)$ (likewise $\beta_k(s)$) is much larger for ISI channels. Consequently, the $\overset{*}{\log}$ operator is less sensitive to smaller metrics that may be discarded by the M-algorithm in the case of an ISI channel. This explains the anomalous behavior of the M-BCJR algorithm.

CHAPTER V

CONVERGENCE BEHAVIOR OF M-BCJR EQUALIZER

The EXIT charts of an iterative decoder help us in understanding its error performance and also in the prediction of the convergence *threshold* E_b/N_o. The threshold of convergence is defined as the smallest E_b/N_o at which the decoding trajectory in the EXIT chart just manages to sneak through a narrow tunnel; the rate of convergence toward a low BER is rather slow but possible as the transfer characteristics of the component decoders no longer intersect. We shall henceforth abbreviate the threshold E_b/N_o predicted from EXIT charts as SNR_{EX} and the actual threshold E_b/N_o observed from BER simulations as SNR_{AC}.

Our simulations showed that for IED using the BCJR equalizer $SNR_{EX} \approx SNR_{AC}$. However, for IED using the M-BCJR equalizer on precoded channels, $SNR_{AC} > SNR_{EX}$. In the case of non-precoded channels, the concept of convergence threshold is not applicable because the EXIT charts of the equalizer and the outer decoder always intersect. Therefore, we shall focus only on the convergence behavior of the M-BCJR equalizer on precoded channels.

A. Convergence Behavior of BCJR Equalizer

From the EXIT charts of IED using the BCJR equalizer on the $1 \oplus D$ precoded $[0.5, 0.5, 0.5, 0.5]$ channel shown in Figure 14, we can see that its expected convergence threshold is $SNR_{EX} = 3.9$ dB. The actual asymptotic performance of IED with the BCJR equalizer on this channel is plotted in Figure 21. An information block length of N=87,500 and 15 iterations were used. At $E_b/N_o = 4.0$ dB, the BER is 10^{-6} or better when more than 10 iterations are used. In the region $3.8 < E_b/N_o < 4.0$, we observe a rapid improvement in BER. Therefore we conclude that the IED actually

converges at E_b/N_o=3.9 dB *i.e.,* SNR_{AC}=SNR_{EX}=3.9 dB.

The EXIT charts of the BCJR equalizer at E_b/N_o = 3.9 dB and the outer

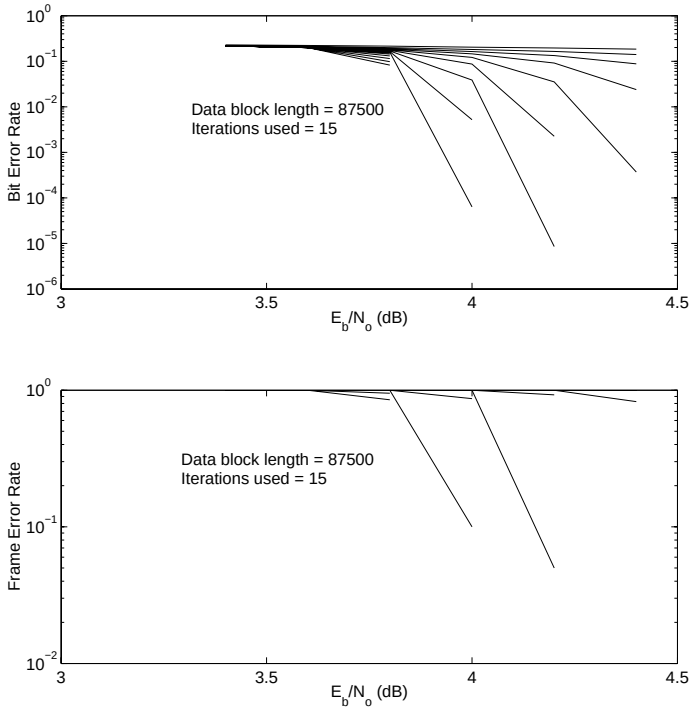

Fig. 21. BCJR performance: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

$[1, 23/35]_8$ decoder are plotted in Figure 22. We used respective block lengths of 250,000 and 100,000 to plot these EXIT charts. It also illustrates the actual decoding trajectory of the IED at E_b/N_o = 3.9 dB obtained by averaging over 25 blocks of information length N=100,000 bits. It just manages to squeeze through the *bottle-neck region* and slowly converges to a low BER.

B. Convergence Behavior of M-BCJR Equalizer

The EXIT charts for M=5 and M=3 equalizers for the precoded $[0.5, 0.5, 0.5, 0.5]$ channel are also plotted in Figure 14. The precoder used was $1 \oplus D$. Based on

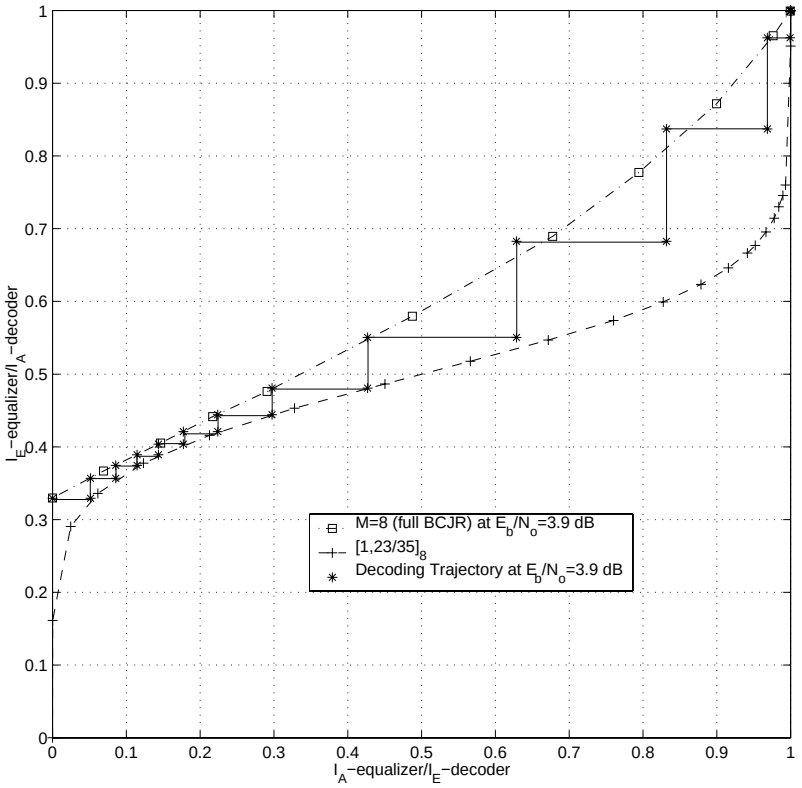

Fig. 22. BCJR decoding trajectory: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

these EXIT charts, we may predict an $SNR_{EX} = 4.0$ dB for the M=5 BCJR and an $SNR_{EX} = 4.4$ dB for the M=3 BCJR algorithm. The IED simulation results using these respective equalizers are plotted in Figures 23 and 24. These performance curves were obtained using 15 iterations and information block lengths of N=87,500 and N=100,000 respectively.

As Figure 23 shows, the IED using M=5 equalizer decodes without any errors at

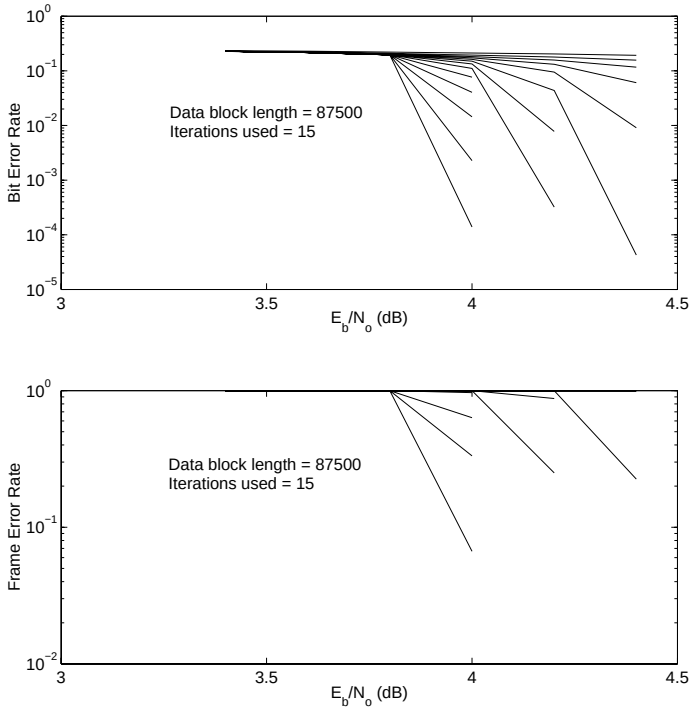

Fig. 23. M=5 BCJR performance: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

$E_b/N_o = 4.0$ dB using 14 or more iterations. Hence the actual convergence threshold is $SNR_{AC} = 4.0$ dB which means $SNR_{AC} = SNR_{EX}$ for this equalizer as well.

Figure 24 shows that IED using M=3 BCJR equalizer provides a low BER at $E_b/N_o = 5.2$ dB after 8 iterations. The performance curves exhibit rapid BER improvement in the *bottle-neck region* which may be roughly specified as $5.0 < E_b/N_o < 5.2$.

Hence we may conclude that $SNR_{AC} \approx 5.1$ dB *i.e.,* $SNR_{AC} > SNR_{EX}$. We will explain the cause of this discrepancy of 0.7 dB between the predicted and actual convergence thresholds in Section C. The decoding trajectories at $SNR_{EX} = 4.4$ dB and $SNR_{AC} = 5.1$ dB are plotted in Figure 25. The M=3 equalizer EXIT chart at E_b/N_o= 4.4 dB and the decoder EXIT chart are also plotted. At SNR_{EX} = 4.4 dB, the decoding trajectory encounters a fixed point and further iterations do not improve the performance. However, at SNR_{AC} = 5.1 dB the decoding trajectory emerges through the tunnel between the decoder characteristics and converges to a low BER.

C. Extrinsic Outputs of M-BCJR Equalizer Not True LLR's

The extrinsic outputs of the M-algorithm are not true LLR's for two reasons.

1. The M-algorithm discards some low-probability paths at each time instant and thus ignores their contributions in all future computations.

2. In the original DTTF trellis, the number of $+1$ and -1 information bit edges is the same. However, in the LLR's computation of the M-algorithm, at any time instant k, the number of surviving $+1$ and -1 information bit edges entering equations (3.2) and (3.3) may not be equal. In the worst case, there may not be any $+1$ and -1 information bit survivor edges at a particular time instant. Therefore, they may not be called as LLR's in its true sense.

In the remainder of this section, we use EXIT charts to assert that the extrinsic outputs of the M-BCJR equalizer are not true LLR's. In subsection 1, we derive a simplified expression for the computation of the mutual informations $I(A;X)$ and $I(E;X)$ of a SISO module assuming that the LLR values are true. Here, X is the information bits stream, A are the input *a priori* LLR values, and E are the output extrinsic values.

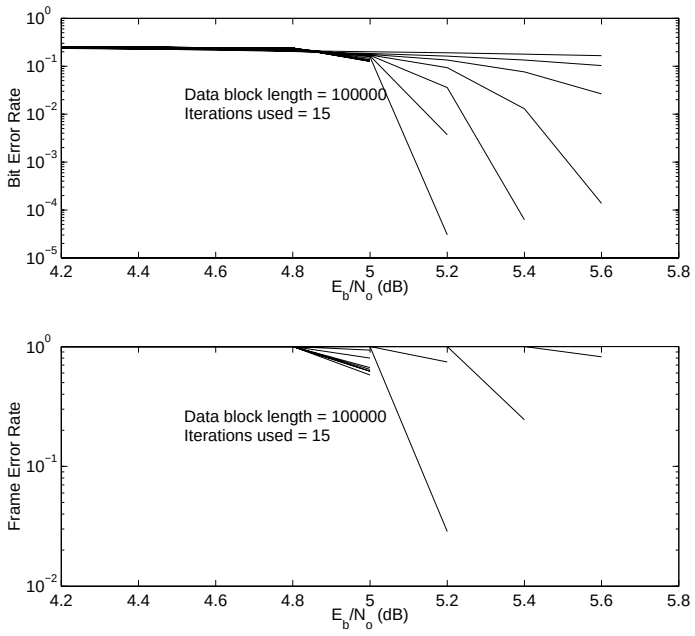

Fig. 24. M=3 BCJR performance: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

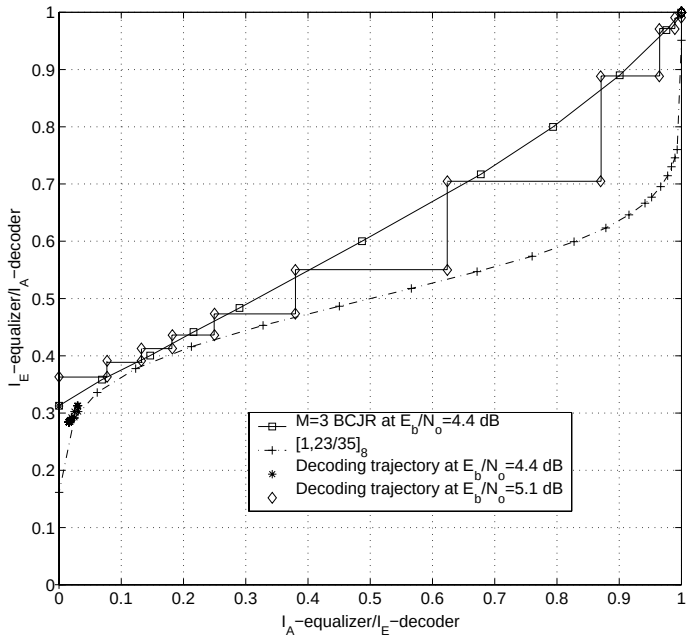

Fig. 25. M=3 BCJR decoding trajectories: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

1. Simplified Calculation of $I(A;X)$ and $I(E;X)$ for True LLR's

In plotting EXIT charts, the mutual informations $I(A;X)$ and $I(E;X)$ of a SISO module are computed from equations (3.20) and (3.22) respectively. They use the conditional distributions $p_A(\xi|X=x)$ and $p_E(\xi|X=x)$,$X \in \chi$ of the *a priori* and extrinsic output LLR's respectively for each symbol in the input alphabet χ of the SISO device. For our purposes, the input alphabet is $\chi = \{-1,+1\}$ always.

Let $p_Z(\xi|X=x),\ X \in \{-1,+1\}$ be the conditional distributions of the random variable Z. Assume that Z represents the true LLR of the random variable X defined as

$$L_Z(x) = \log\left\{\frac{p(X=+1|\xi)}{p(X=-1|\xi)}\right\} = \log\left(\frac{p}{1-p}\right) \tag{5.1}$$

where p is the conditional probability that $X=+1$. Using Bayes' theorem, equation (5.1) may be rewritten as

$$L_Z(x) = \log\left(\frac{p}{1-p}\right) = \log\left\{\frac{p_Z(\xi|X=+1)}{p_Z(\xi|X=-1)}\right\} \tag{5.2}$$

because $Pr(X=-1)=Pr(X=+1)=1/2$. The mutual information $I(Z;X)$ computes as

$$I_Z = \frac{1}{2}\cdot\sum_{x=-1,+1}\int_{-\infty}^{+\infty} p_Z(\xi|X=x) \times \log_2\left\{\frac{2\cdot p_Z(\xi|X=x)}{p_Z(\xi|X=-1)+p_Z(\xi|X=+1)}\right\} d\xi \tag{5.3}$$

$$0 \le I_Z \le 1. \tag{5.4}$$

Equation (5.3) has the same form as equations (3.20) and (3.22) used for computing $I(A;X)$ and $I(E;X)$ respectively. It may be simplified as

$$I_Z = \frac{1}{2} \cdot \sum_{x=-1,+1} \int_{-\infty}^{+\infty} p_Z(\xi|X=x) \times \left[1 + \log_2 \left\{ \frac{p_Z(\xi|X=x)}{p_Z(\xi|X=-1) + p_Z(\xi|X=+1)} \right\}\right] d\xi \tag{5.5}$$

which expands to

$$I_Z = 1 + \frac{1}{2} \cdot \int_{-\infty}^{+\infty} p_Z(\xi|X=-1) \cdot \log_2 \left\{ \frac{p_Z(\xi|X=-1)}{p_Z(\xi|X=-1) + p_Z(\xi|X=+1)} \right\} d\xi + \frac{1}{2} \cdot \int_{-\infty}^{+\infty} p_Z(\xi|X=+1) \cdot \log_2 \left\{ \frac{p_Z(\xi|X=+1)}{p_Z(\xi|X=-1) + p_Z(\xi|X=+1)} \right\} d\xi. \tag{5.6}$$

Rewriting equation (5.6), we get

$$I_Z = 1 + \frac{1}{2} \cdot \int_{-\infty}^{+\infty} p_Z(\xi|X=-1) \cdot \log_2 \left\{ \frac{1}{1 + \frac{p_Z(\xi|X=+1)}{p_Z(\xi|X=-1)}} \right\} d\xi + \frac{1}{2} \cdot \int_{-\infty}^{+\infty} p_Z(\xi|X=+1) \cdot \log_2 \left\{ \frac{\frac{p_Z(\xi|X=+1)}{p_Z(\xi|X=-1)}}{1 + \frac{p_Z(\xi|X=+1)}{p_Z(\xi|X=-1)}} \right\} d\xi. \tag{5.7}$$

Using equation (5.2) and the fact that $Pr(X=-1) = Pr(X=+1) = 1/2$, equation (5.7) becomes

$$I_Z = 1 + \int_{-\infty}^{+\infty} p_Z(\xi|X=-1) \cdot Pr(X=-1) \cdot \log_2(1-p) d\xi + \int_{-\infty}^{+\infty} p_Z(\xi|X=+1) \cdot Pr(X=+1) \cdot \log_2(p) d\xi \tag{5.8}$$

which may be rewritten using Bayes' theorem and equation (5.1) as

$$I_Z = 1 + \int_{-\infty}^{+\infty} \left[p_Z(\xi) \cdot \{(1-p) \cdot \log_2(1-p) + p \cdot \log_2(p)\} \right] d\xi. \tag{5.9}$$

By replacing the continuous integration by a summation operation over the data block in equation (5.9), the expression for calculation of mutual information may be

equivalently written as

$$I_Z = 1 + \frac{1}{N} \cdot \sum_{n=0}^{N-1} \{(1-p_n)\cdot\log_2(1-p_n) + p_n\cdot\log_2(p_n)\}\,. \tag{5.10}$$

Equation (5.10) is simple to implement and has lower computational complexity than equations (3.20) and (3.22). It also does not require the computation of the conditional distributions $p_Z(\xi|X=x),\ X\in\{-1,+1\}$.

2. EXIT Charts Using the Simplified Expression

In plotting EXIT charts, the input *a priori* LLR's are randomly generated by imposing certain conditions as described in Chapter IV. Hence they always behave as true LLR's. However, the extrinsic outputs of a SISO module may not be true LLR's. Given this scenario, an EXIT chart plotted using equation (5.10) will not concur with its original EXIT chart if the extrinsic outputs are not true LLR's.

The EXIT charts for the precoded $[0.5, 0.5, 0.5, 0.5]$ and $[0.5, 0.5, 0.0, 0.5, 0.5]$ channels obtained using equation (5.10) for the computation of mutual informations are shown in Figures 26 and 27. The original EXIT charts for these respective channels were plotted in Figures 14 and 15. The EXIT charts for the BCJR equalizer plotted using the equation (5.10) and the original equations (3.20) and (3.22) for mutual information are perfectly coincident for both the DTTF models. This is because the extrinsic outputs of the BCJR equalizer are true LLR's. On the other hand, the EXIT charts for the M-BCJR equalizers do not concur on either of these channels. In fact, similar results were observed on several other channels. This proves that the extrinsic outputs of an M-BCJR equalizer are not true LLR's. In addition, the new EXIT charts predict that the performance of the M-BCJR equalizer is superior to its actual performance. Therefore, we may also conclude that the extrinsic outputs of the M-BCJR equalizer are optimistic.

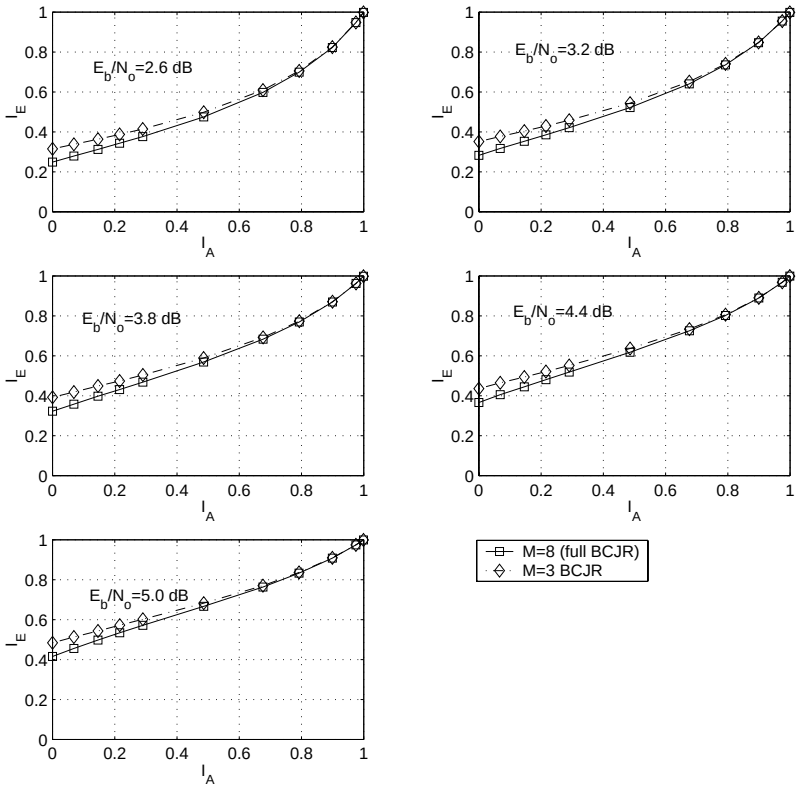

Fig. 26. EXIT charts using eqn. (5.10): 1⊕D precoded $[0.5, 0.5, 0.5, 0.5]$

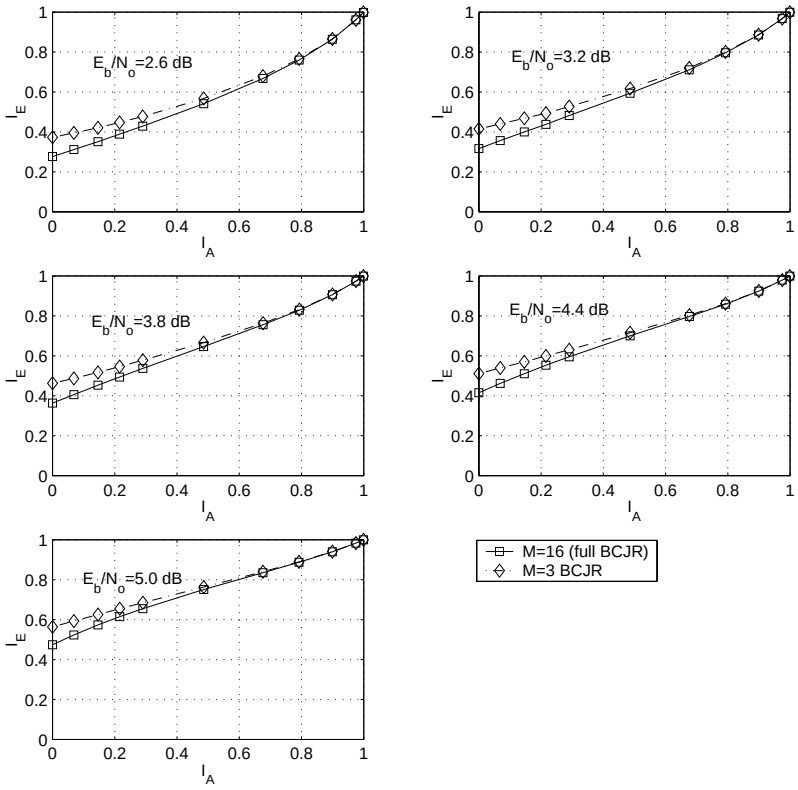

Fig. 27. EXIT charts using eqn. (5.10): 1⊕D precoded $[0.5, 0.5, 0.0, 0.5, 0.5]$

D. Genie-Assisted IED

Since the extrinsic outputs of an M-BCJR equalizer are not true LLR'sthe traditional practice of directly using them as feedback to the outer decoder is suboptimal. The EXIT charts of an IED device however, project its best achievable performance. In other words, they are drawn with the intrinsic assumption that the extrinsic outputs of the component SISO modules are used in an optimal fashion. In plotting EXIT charts, the mutual informations $I(A;X)$ and $I(E;X)$ are computed from equations (3.20) and (3.22) using their respective conditional distributions $p_A(\xi|X=x)$ and $p_E(\xi|X=x)$,$X\in\{-1,+1\}$.

The above observations collectively suggest that the true LLR's at the output of the M-BCJR equalizer may be estimated using Bayes' theorem if the conditional distributions $p_E(\xi|X=x),\ X\in\{-1,+1\}$ are known. Consequently, the actual error performance threshold SNR_{AC} of an M-BCJR equalizer may converge to the predicted threshold SNR_{EX}.

Using Bayes' theorem and the conditional pdf's $p_E(\xi|X=-1)$ and $p_E(\xi|X=+1)$, the true LLR's at the output of an M-BCJR equalizer are estimated from its extrinsic outputs as

$$L_E(x_k)=\frac{Pr(X=+1|\xi)}{Pr(X=-1|\xi)}=\frac{P_E(\xi|X=+1)}{P_E(\xi|X=-1)} \tag{5.11}$$

because $Pr(X=-1)=Pr(X=+1)=1/2$.

It should be noted that the estimation of true LLR's using equation (5.11) is not possible because of the impracticability of knowing the conditional distributions $p_E(\xi|X=-1)$ and $p_E(\xi|X=+1)$. For this reason, the IED in which the true LLR's are estimated from equation (5.11) is called as a *genie-assisted* IED.

In the following, we present some EXIT charts of M-BCJR equalizers and simu-

lated decoding trajectories on different channels to illustrate the effectiveness of the *genie-assisted* IED. Figure 28 shows the EXIT charts of the M=3 BCJR equalizer for the $1 \oplus D$ precoded $[0.5, 0.5, 0.5, 0.5]$, $[1, 23/35]_8$ outer decoder and the decoding trajectories of traditional and *genie-assisted* IED's. By traditional IED, we mean the IED which directly uses the extrinsic outputs of the M-BCJR equalizer as feedback to the outer decoder. These decoding trajectories were simulated using 20 iterations and an information block length of N=100,000. From our convergence analysis of the M-BCJR equalizer on the precoded $[0.5, 0.5, 0.5, 0.5]$ channel in Section B, we know that the traditional IED for this channel has a threshold $SNR_{AC} = 5.1$ dB (Figure 25). The IED predicted threshold is $SNR_{EX} = 4.4$ dB at which the *genie-assisted* IED converges as seen in Figure 28. This confirms that the true LLR's can be accurately estimated using equation (5.11).

We further present a few more EXIT charts and decoding trajectories to verify the performance of the *genie-assisted* IED. The EXIT charts for IED on $1 \oplus D$ precoded $[0.5, 0.5, 0.5, 0.5]$ with $[1, 1/3]_8$ as the outer decoder are plotted in Figure 29. They were obtained using data blocks of length N=250,000. It is a well known result that a smaller constraint length outer convolutional code allows the IED to converge at a smaller SNR_{AC}. Due to this reason, we chose the outer code as $[1, 1/3]_8$. From the EXIT charts, we predict $SNR_{EX} = 3.2$ dB for the M=3 BCJR equalizer.

The error performance of the traditional IED using the M=3 BCJR equalizer converges at $SNR_{AC} = 3.35$ dB. Its decoding trajectory is shown in Figure 30. The decoding trajectories of the traditional and *genie-assisted* IED's at $E_b/N_o = 3.2$ dB are illustrated in Figure 31. All these decoding trajectories were simulated using an information block length of N=250,000 and 40 iterations. These plots show that the *genie-assisted* IED converges at $SNR_{EX} = 3.2$ dB. In Figure 31, the traditional IED at $SNR_{EX} = 3.2$ dB did not encounter a fixed point. It might slowly squeeze through

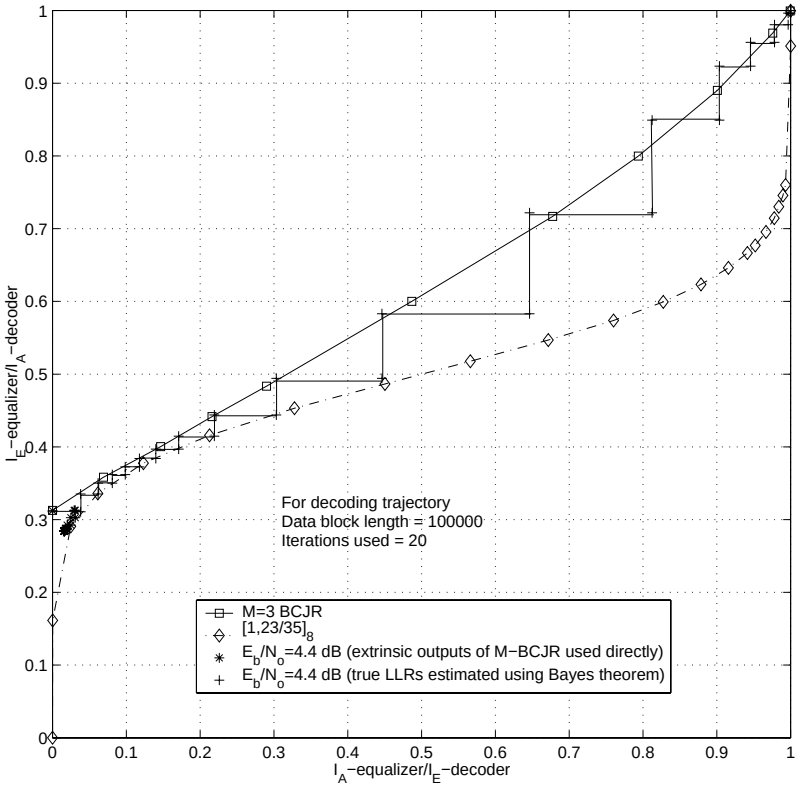

Fig. 28. Genie-assisted IED: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{23}{35}]_8$

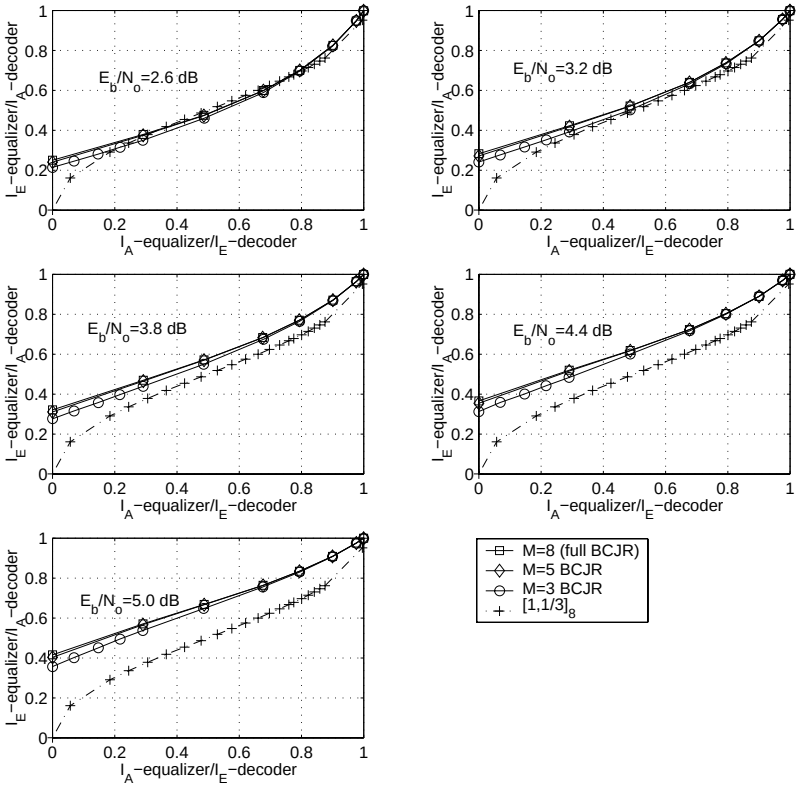

Fig. 29. EXIT charts: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{1}{3}]_8$

the *bottle-neck region* between the transfer characteristics of the M=3 BCJR and the outer decoder and eventually converge to a low BER. But, the number of iterations required to achieve an acceptable BER of 10^{-5} or better could be as high as 100.

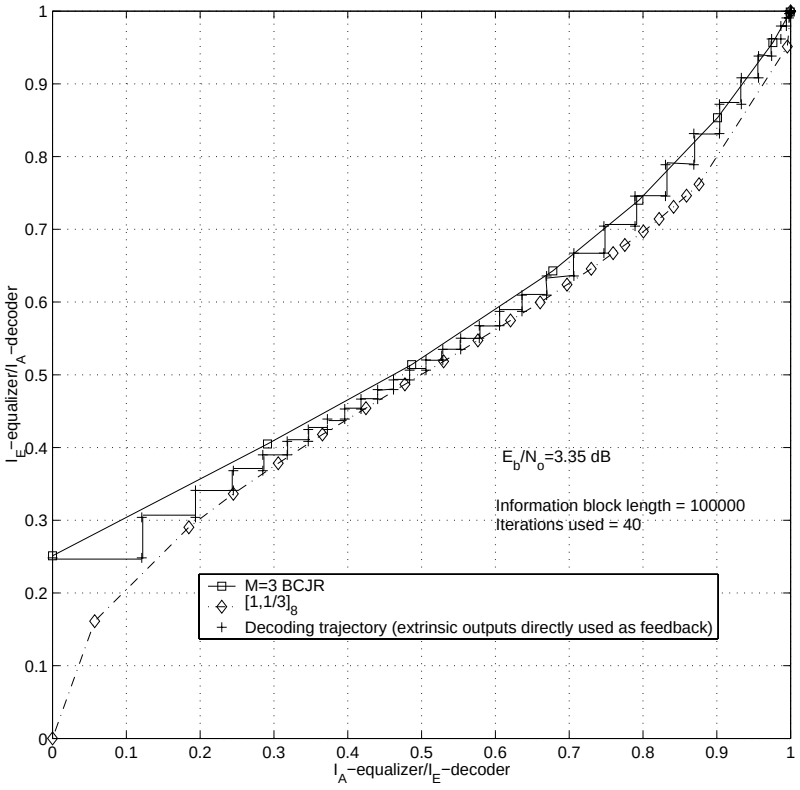

Fig. 30. Traditional IED: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{1}{3}]_8$

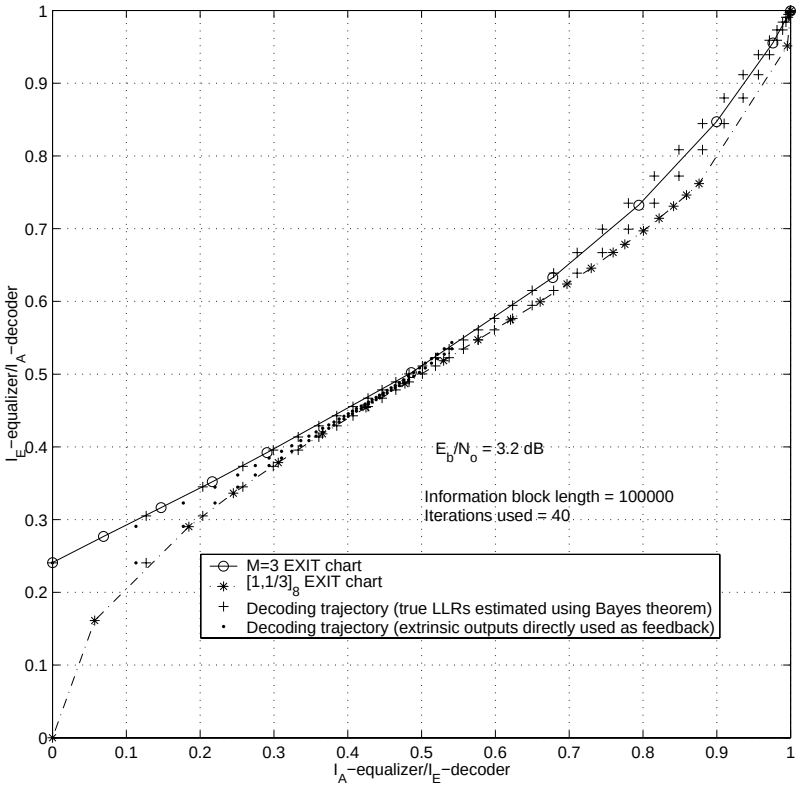

Fig. 31. Traditional and genie IED's: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5], [1, \frac{1}{3}]_8$

CHAPTER VI

ESTIMATION OF TRUE LLR'S FROM M-BCJR EQUALIZER OUTPUTS

The true LLR's at the output of an M-BCJR equalizer can be estimated perfectly if the conditional distributions $p_E(\xi|X=x),\ X \in \{-1,+1\}$ of the output extrinsic values are known. Since this is an impractical requirement, we design a practical and reliable estimator based on some approximations.

It may be noted that the concept of refining the extrinsic outputs of a suboptimal SISO device before they are used as *a priori* input to the other SISO device in an iterative decoder is not entirely new. For example, the extrinsic outputs of the SOVA are optimistic. They are brought closer to the true LLR's by normalization [25] which consists of multiplying every extrinsic output by a constant factor that depends on the current BER. This normalization constant is computed by assuming a Gaussian distribution for the extrinsic outputs. In our initial efforts to design a reliable estimator, we followed a similar approach.

A. Estimation Using Gaussian Approximation

The extrinsic outputs of the M-BCJR equalizer are assumed to be Gaussian distributed. Unlike simple convolutional codes, the ISI channels behave as non-linear codes. In this scenario, we verified the validity of our assumption from simulations as a precautionary measure. The actual distribution of the extrinsic outputs matched closely with the Gaussian assumption but we discovered some interesting facts about the behavior of the M-BCJR equalizer. On non-precoded channels, the conditional distributions of the extrinsic outputs $p_E(\xi|X=x),\ X \in \{-1,+1\}$ were quite similar in shape and had similar means and variances as well. On the other hand, the conditional distributions of the extrinsic outputs $p_E(\xi|X=x),\ X \in \{-1,+1\}$ on precoded channels

were quite dissimilar. As an example, the histograms of $p_E(\xi|X=x),\ X\in\{-1,+1\}$ for M=3 BCJR on the precoded $[0.5, 0.5, 0.5, 0.5]$ channel are shown in Figure 32. These results suggest that the M-BCJR equalizer offers unequal protection to the information bits -1 and $+1$ when the channel is precoded. Therefore, the problem of estimating the true LLR's assumes different forms for precoded and non-precoded channels. However, for reasons explained earlier, we try to design the estimator for the precoded case only.

In actual decoding, it is only practical to compute the mean μ_D and variance

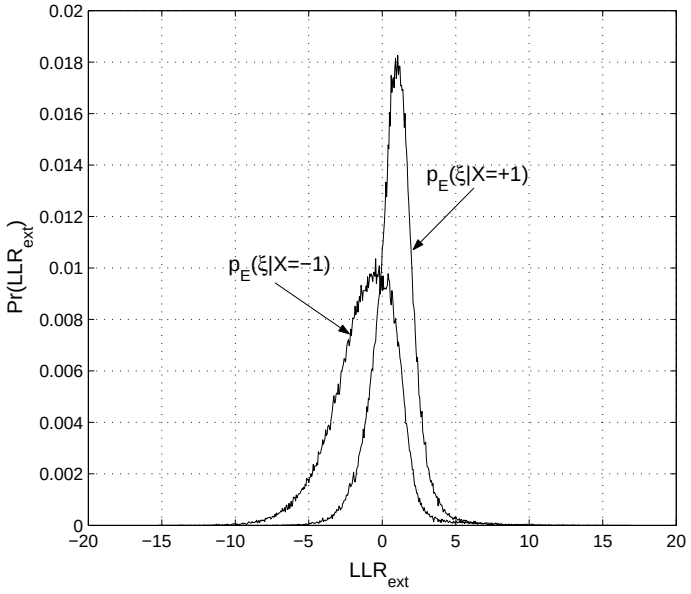

Fig. 32. Conditional pdf's of M=3 BCJR: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5]$

σ_D^2 of all the extrinsic outputs of each block. It is not possible to compute the means $\mu_{D_{-1}}$, $\mu_{D_{+1}}$ and the variances $\sigma^2_{D_{-1}}$, $\sigma^2_{D_{+1}}$ of the conditional distributions $p_E(\xi|X=-1)$ and $p_E(\xi|X=+1)$ from the extrinsic outputs themselves. We overcome this problem by using lookup tables of the means μ_W, μ_{-1}, μ_{+1} and variances σ_W^2, σ_{-1}^2, σ_{+1}^2 at several points along the EXIT chart. The subscript W refers to the extrinsic out-

puts as a whole and the subscripts -1 and $+1$ refer to the corresponding conditional distributions. These tables are generated from simulations by averaging over several blocks assuming that the transmitted bits X are known.

During actual decoding, we compute the whole mean μ_D and variance σ_D^2 and estimate the means $\mu_{D_{-1}}$, $\mu_{D_{+1}}$ and variances $\sigma_{D_{-1}}^2$, $\sigma_{D_{+1}}^2$ by interpolation. The true LLR's are then computed using

$$L_E(x_k) = \log\left(\frac{\sigma_{D_{-1}}}{\sigma_{D_{+1}}}\right) - \frac{1}{2}\left[\frac{(\xi-\mu_{D_{+1}})^2}{\sigma_{D_{+1}}^2} - \frac{(\xi-\mu_{D_{-1}})^2}{\sigma_{D_{-1}}^2}\right] \tag{6.1}$$

where ξ is the extrinsic output.

Our simulations using equation (6.1) for the estimation of true LLR's did not improve the error performance of the turbo equalizer. This shows that the above true LLR estimator is unreliable. The ineffectiveness of estimator (6.1) may be attributed to the heavy tails of the conditional distributions $p_E(\xi|X=x),\ X\in\{-1,+1\}$ in the direction of the opposite information symbol.

B. Estimation Using Stored Histograms

This approach is intrinsically similar to the Gaussian approximation method described in Section A. However, the assumption of Gaussian distribution for the extrinsic outputs is discarded. Instead, a string of histograms $h_E^m(\xi|X=x),\ X\in\{-1,+1\},\ m=1,2,\ldots,T$ where T is the total number of iterations, are used to estimate the true LLR's. They are computed from the *genie-assisted* IED by averaging the true conditional distributions of the M-BCJR equalizer extrinsic outputs $p_E(\xi|X=x),\ X\in\{-1,+1\}$ at the end of iteration m over 25 blocks. These histograms are specific to a particular E_b/N_o and may be used only when operating at that E_b/N_o. Therefore, it is important to have *a priori* knowledge of the channel

signal-to-noise ratio in order to use this method.

In actual performance simulations using these stored histograms, the true LLR's at the end of iteration m are estimated using

$$L_E(x_k) = \frac{Pr(X=+1|\xi)}{Pr(X=-1|\xi)} = \frac{h_E^m(\xi|X=+1)}{h_E^m(\xi|X=-1)} \tag{6.2}$$

which is similar to equation (5.11). The performance of IED using stored histograms to estimate the true LLR's from equation (6.2) at the M=3 BCJR equalizer output on the precoded $[0.5, 0.5, 0.5, 0.5]$ channel is shown in Figure 33. The outer code was

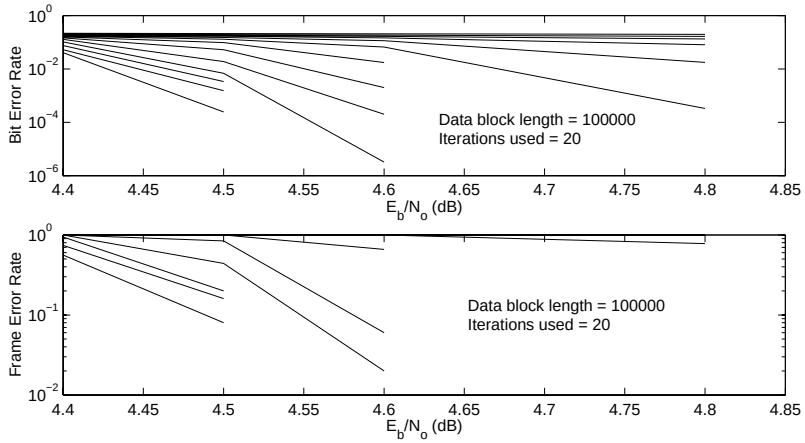

Fig. 33. Performance of M=3 BCJR: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5]$. True LLR's at the M-BCJR equalizer output estimated using stored histograms

$[1, 23/35]_8$ and the precoder was $1 \oplus D$. An information block length of N=100,000 and 20 iterations were used for the simulations. At $E_b/N_o = 4.4$ dB, IED failed to evolve. However, at $E_b/N_o = 4.5$ dB, there were no errors at the end of 13 iterations. As seen in Chapter V, the expected convergence threshold for the above code is $SNR_{EX} = 4.4$ dB and the observed threshold is $SNR_{AC} = 5.1$ dB. This shows that this method of estimating true LLR's from stored histograms significantly improves the error performance of IED using M-BCJR equalizers on precoded channels. In Figure

34, we plotted the decoding trajectory of IED using stored histograms at E_b/N_o=4.5 dB and the EXIT charts of the M=3 equalizer and the outer decoder. The plot shows the efficacy of this true LLR's estimator.

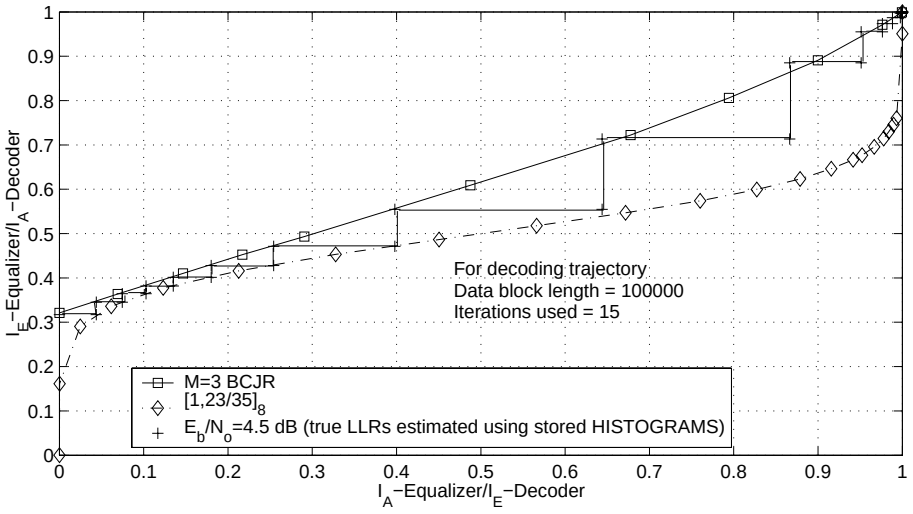

Fig. 34. EXIT chart of M=3 BCJR: 1⊕D precoded $[0.5, 0.5, 0.5, 0.5]$. True LLR's at the M-BCJR equalizer output estimated using stored histograms

CHAPTER VII

CONCLUSIONS

In this thesis, we focussed on understanding the behavior of the M-BCJR [7] equalizer in IED and also compared its performance with other low complexity equalizers. We studied the performance of the M-BCJR equalizer on precoded and non-precoded channels with the help of BER, FER simulations and EXIT charts [8]. The performance degradation of the M-BCJR equalizer as compared with the BCJR equalizer was found to be rather small. We also compared the losses in achievable channel capacity of IED suffered when the BCJR equalizer is replaced with the M-BCJR equalizer and an MMSE LE. It was observed that the capacity loss for the M-BCJR equalizer was very small. On the other hand, the MMSE LE suffered a substantial loss in channel capacity. It was also observed that precoding only affects the IED performance. It does not affect the channel capacity.

Simulation results showed that the M-BCJR algorithm suffers significant losses in the case of simple convolutional decoders. This contrasting behavior of the M-BCJR algorithm in the cases of ISI channels and convolutional codes was explained by showing that the metrics computed during the forward and backward recursions of the M-BCJR equalizer have much larger variance than in the case of a convolutional code. The larger variance of the metrics in the M-BCJR equalizer makes the algorithm less sensitive to the paths discarded.

Next, the convergence behavior of the M-BCJR equalizer was studied and compared with the BCJR equalizer. In contrast with the BCJR equalizer, the actual convergence threshold of the M-BCJR equalizer was found to disagree with its threshold predicted from the EXIT charts. These observations were justified by showing that the extrinsic outputs of the M-BCJR equalizer are not true LLR's. It was there-

fore argued that these values cannot be used directly as *a priori* input to the outer decoder. We showed that the true LLR's can be estimated using Bayes' theorem if the conditional distributions of the extrinsic outputs for each symbol in the input alphabet of the ISI channel are known. We also presented some EXIT charts and decoding trajectories for such a *genie-assisted* IED using the M-BCJR equalizer to prove its effectiveness.

Since the *genie-assisted* IED assumes perfect knowledge of the actual information bits, it is not practically realizable. Hence a practical and reliable estimator for computing the true LLR's from the extrinsic outputs of the M-BCJR equalizer is required and we tried two different approaches to design such an estimator. In the first method, the extrinsic outputs of the M-BCJR equalizer were assumed to be Gaussian distributed. However, the Gaussian approximation estimator did not yield satisfactory results. The second method consisted of using stored histograms of the conditional distributions of extrinsic outputs computed from simulations. This method was successful and significantly improved the performance of IED using the M-BCJR equalizer.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[2] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier and A. Glavieux, "Iterative correction of inter-symbol interference: Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sept.-Oct. 1995.

[3] S. Benedetto, G. Montorsi, D. Divsalar and F. Pollara, "Serial concatenation of interleaved codes: Design and performance analysis," *IEEE Trans. Info. Theory*, vol. 42, pp. 409–429, Apr. 1998.

[4] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE GLOBECOM*, Dallas, TX, Nov. 1989, pp. 47.1.1–47.1.6.

[5] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[6] K. R. Narayanan, U. Dasgupta and B. Lu, "Low complexity turbo equalization with binary precoding," in *Proc. IEEE Int. Conf. Commun.*, New Orleans, LA, 2000, vol. 1, pp. 1–5.

[7] V. Franz and J. B. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 186–195, Feb. 1998.

[8] S. ten Brink, "Convergence behaviour of iteratively decoded parallel concatenated codes.," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[9] J. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[10] A. Glavieux, C. Laot and J. Labat, "Turbo equalization over a frequency selective channel," in *Proc. Int. Symp. Turbo Codes*, Brest, France, Sept. 1997, pp. 96–102.

[11] D. Raphaeli and A. Saguy, "Linear equalizers for turbo equalization: A new optimization criterion for determining the equalizer taps," in *Proc. 2nd Int. Symp. Turbo Codes*, Brest, France, Sept. 2000, pp. 371–374.

[12] Z. Wu and J. Cioffi, "Turbo decision aided equalization for magnetic recording channels," in *Proc. IEEE GLOBECOM*, Rio de Janeiro, Brazil, Dec. 1999, vol. 1B, pp. 733–738.

[13] M. Tuchler, R. Koetter and A. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, pp. 754–767, May 2002.

[14] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Proc. Int. Symp. Turbo Codes*, Brest, France, Sept. 1997, pp. 1–11.

[15] S. Benedetto, G. Montorsi, D. Divsalar and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding," *IEEE Trans. Info. Theory*, vol. 44, pp. 909–926, May 1998.

[16] S. Haykin, *Communication Systems*, 3rd ed. New York: Wiley, 1994.

[17] D. Raphaeli and Y. Zarai, "Combined turbo equalization and turbo decoding," in *Proc. IEEE GLOBECOM*, Phoenix, AZ, Nov. 1997, vol. 2, pp. 639–643.

[18] A. Anastaspoulos and K. Chugg, "Iterative equalization/decoding for TCM for frequency-selective fading channels," in *Proc. 31st Asilomar Conf. Signals, Systems & Computers*, Pacific Grove, CA, Nov. 1997, vol. 1, pp. 177–181.

[19] A. Berthet, R. Visoz and P. Tortelier, "Sub-optimal turbo-detection for coded 8-PSK signals over ISI channels with applications to EDGE advanced mobile systems," in *Proc. 11th Int. Symp. Personal, Indoor and Mobile Radio Communications*, London, UK, Sept. 2000, vol. 1, pp. 151–157.

[20] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun.*, Seattle, WA, June 1995, pp. 1009–1013.

[21] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linkoping University, Sweden, 1996.

[22] T. J. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[23] A. Ashikhmin, G. Kramer and S. ten Brink, "Code rate and the area under extrinsic information transfer curves," in *Proc. IEEE Int. Symp. Inform. Theory*, Lausanne, Switzerland, June–July 2002, pp. 115–115.

[24] K. R. Narayanan, "Effect of precoding on the convergence of turbo equalization for partial response channels," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 686–698, Apr. 2001.

[25] L. Papke and P. Robertson, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc. IEEE Int. Conf. Commun.*, Dallas,

TX, June 1996, vol. 1, pp. 102–106.

VITA

Raja Venkatesh Tamma was born on October 5, 1979, in India. He received his B.E. (Hons.) degree in electrical and electronics engineering from the Birla Institute of Technology & Science, Pilani in June 2000. He received his M.S. degree in electrical engineering from Texas A&M University, College Station in May 2003. His research interests include coding theory and turbo equalization with application to digital magnetic recording systems. His permanent address is C608, Manish, Evershinenagar, Malad (West), Mumbai-400064, India. His email address is rajatamma@yahoo.com

The typist for this thesis was Raja Tamma.